



Développement d'algorithmes dynamiques et stochastiques pour le problème de transport de patients dans les hôpitaux

Mémoire

Mohamed Zied Torkhani

Maîtrise en sciences de l'administration - avec mémoire
Maître ès sciences (M. Sc.)

Québec, Canada

Développement d'algorithmes dynamiques et stochastiques pour le problème de transport de patients dans les hôpitaux

Mémoire

Mohamed Zied Torkhani

Sous la direction de:

Jean-François Côté, directeur de recherche
Paolo Landa, codirecteur de recherche

Résumé

Ce mémoire traite un problème de transport de personnes dans un contexte hospitalier, connu sous le nom du problème de brancardier. L'objectif est de construire des itinéraires qui répondent aux demandes de transports émergentes entre les différents services d'un grand centre hospitalier en temps réel, en minimisant le temps total de retard pondéré. Ce problème est traité comme un problème de cueillettes et de livraisons multitrajets qui considère des fenêtres de temps souples, une flotte hétérogène de véhicules et des contraintes liées à la capacité. Les requêtes de transport de patients sont imprévisibles et dynamiques. Elles sont révélées lorsqu'un patient nécessite un service de transport pour des raisons médicales. Ce travail présente trois approches de résolution du problème de transport de patients, à noter une première approche statique, une deuxième dynamique et une troisième stochastique. De plus, une stratégie d'attente et deux stratégies de relocalisation de véhicules ont été développées. Les approches sont évaluées sur des données réelles d'un grand hôpital, le Policlinico Sant'Orsola-Malpighi de la mairie de Bologne en Italie.

Mots clés : Transport de patients. Problème de cueillettes et de livraisons dynamique. Recherche adaptative à grand voisinage. Heuristique de réoptimisation. Approche de planification basée sur des scénarios échantillonnés. Fonction de consensus.

Abstract

The following study presents the problem of transportation of patients in the medical field. Demand in this context is unpredictable and revealed dynamically. The objective is to develop an algorithm capable of constructing efficient and effective routes in real time while minimizing the total weighted lateness. This problem is considered as a multitrip pickup and delivery problem with soft time windows, heterogeneous fleet, and capacity constraints. This work presents a detailed description of the discussed problem and proposes three approaches to solve it: a static approach, a dynamic approach and a stochastic one. Moreover, it presents a waiting and two relocalisation strategies. These approaches have all been tested and evaluated using real data collected from the medical campus of Policlinico Sant'Orsola-Malpighi of the town Hall of Bologna in Italy.

Key words: Transportation of patients. Dynamic pickup and delivery problem. Adaptive large neighbourhood search. Reoptimization heuristic. Scenario-based planning approach. Consensus function

Table des matières

Résumé	ii
Abstract	iii
Table des matières	iv
Liste des tableaux	vi
Liste des figures	vii
Liste des sigles et abréviations	viii
Remerciements	ix
Introduction	1
1 Revue de la littérature	4
1.1 Les problèmes de tournées de véhicules	4
1.2 Les problèmes de cueillettes et de livraisons	5
1.3 Les problèmes de transport à la demande	7
1.4 Le problème des brancardiers	9
1.5 Méthodes de résolution	12
1.6 Contribution à la recherche	17
2 Le problème des brancardiers	19
2.1 Problématique	19
2.2 Hypothèses simplificatrices	20
2.3 Analyse de la situation actuelle	21
2.4 Analyse des données	22
2.5 Modélisation	26
3 Algorithmes et approches de résolution	29
3.1 Méthodologie de résolution	29
3.2 Approche statique	31
3.3 Approche dynamique	36
3.4 Approche stochastique	40
4 Étude expérimentale	44
4.1 Conception des instances tests	44

4.2 Paramètres	45
4.3 Présentation des résultats finaux	47
4.4 Analyse de sensibilité	56
Conclusion	59
Bibliographie	61

Liste des tableaux

2.1	Les pairs de nœuds les plus visités pour chaque jour de la semaine	25
4.1	Tester différentes itérations de l'ALNS	46
4.2	Tester différents nombres de scénario	46
4.3	Résultats de l'approche statique	48
4.4	Résultats de l'approche dynamique	49
4.5	Analyse de la deuxième stratégie	51
4.6	Analyse de la troisième stratégie	51
4.7	Résultats de l'approche stochastique	52
4.8	Les résultats des différents algorithmes	53
4.9	Les statistiques du test de normalité de Shapiro-Wilk	54
4.10	Les valeurs des statistiques	56
4.11	L'impact du changement de nombre de véhicules	58

Liste des figures

2.1	Plan de l'hôpital universitaire Policlinico Sant'Orsola-Malpighi.	20
2.2	Décomposition de la série chronologique des requêtes de transport reçues au cours de l'année 2016.	23
2.3	Nombres de requêtes moyens par jour.	23
2.4	Tendance annuelle des requêtes de transport pour chaque jour de la semaine. .	24
2.5	Fréquence de visite des origines et des destinations.	25
2.6	Nombres de requêtes moyens par heures.	26
4.1	Histogramme des retards pour chaque approche	55

Liste des sigles et abréviations

ALNS	Recherche adaptative à grand voisinage (Adaptive Large Neighbourhood Search)
B&R	Séparation et regret (Branch and Regret)
DARP	Problème de transport à la demande (Dial-A-Ride-Problem)
DSHH	Heuristique de couverture dynamique et stochastique (Dynamic and Stochastic Hedging Heuristic)
MSA	Approche à scénarios multiples (Multiple-Scenario Approach)
PDP	Problème de cueillettes et de livraisons (Pick-up and Delivery Problem)
PIB	Produit intérieur brut
SBPA	Approche de planification basée sur des scénarios (Scenario-Based Planning Approach)
SDDP	Problème de livraison le jour même (Same Day Delivery Problem)
SDVRP	Problème de tournées de véhicules stochastique et dynamique (Stochastic and Dynamic Vehicle Routing Problem)
TSP	Problème du voyageur de commerce (Traveling Salesman Problem)
VRP	Problème de tournées de véhicules (Vehicle Routing Problem)

Remerciements

Premièrement, je tiens à remercier mes deux directeurs de recherche, Pr. Jean-François Côté et Pr. Paolo Landa, qui m'ont encadré et orienté dans la réalisation de ce travail. Je les remercie pour leur présence, leur patience, leur disponibilité et leur partage de savoir tout au long de ce projet.

Je tiens également à remercier tout le corps professoral de la faculté des sciences d'administration (FSA) et du Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT) pour toutes leurs contributions directs et indirects et leur partage de connaissances.

Finalement, je tiens à remercier toute personne ayant contribué de près ou de loin à la réussite de ce travail.

Introduction

Le secteur de santé est sans doute l'un des secteurs les plus vitaux et essentiels au bien être des nations. En termes de développement humain, ce secteur représente l'un des axes les plus importants de l'index IDH, qui est un indicateur de développement humain (UNPD, 2020). Les dépenses de ce secteur comptent mondialement pour près de 9.58% du PIB de tous les pays, et de 10.79% du PIB canadien pour l'Année 2018 (world bank, 2018). Au Québec par exemple, les dépenses en domaine de santé représentent près de 43% des dépenses de la province, soit 45 milliards CAD en 2020 comparés à 22 milliards en 2004 (Laforest, 2021).

La récente crise sanitaire de la Covid-19 a également prouvé l'importance vitale de ce domaine, et surtout de mettre le point sur la précarité de ses ressources face à une telle situation imprévisible. Ainsi, l'importance d'une meilleure gestion de ces ressources très limitées s'annonce plus que jamais cruciale.

Malgré toutes cette importance et les budgets énormes alloués à son fonctionnement, il est l'un des secteurs dont la logistique n'est pas très bien gérée. Cette crise sanitaire a poussé tout le monde à considérer l'amélioration et l'optimisation de la logistique interne dans les différents établissements de santé. En général, les activités logistiques sont coûteuses et représentent près de 46% des coûts de fonctionnement total d'un hôpital (Landry et Philippe, 2004). Dans les grands hôpitaux, le transport interne occupe une partie importante de ses activités. En fait, pour que les patients reçoivent les soins nécessaires, une centaine de demandes de transport entre les différents services sont effectuées chaque jour.

Dans les grands établissements de santé qui contiennent différents pavillons, la rapidité et la ponctualité du service de transport de patients ont un impact direct sur l'efficacité des services médicaux. Quoique la gestion de ce service s'apparente à être facile, les décisions et les opérations de ce dernier sont critiques, entraînent plusieurs coûts et affectent l'utilisation et la disponibilité des ressources rares comme les salles opératoires. À titre d'exemple, le retard de transport d'un patient à leurs destinations peut affecter les calendriers des médecins en décalant d'autres rendez-vous prévus pour le reste la journée. Dans les cas urgents, un retard important peut même coûter la vie du patient.

Généralement, les opérations de transport interhospitalier sont assurées par les services de la

brancarderie. Cette dernière reçoit des requêtes de transport de patients des différents services hospitaliers, et décide comment affecter sa flotte pour répondre aux différentes demandes tout en respectant les délais et l'urgence des ces dernières. L'affectation se fait en fonction de la disponibilité des véhicules au moment de l'arrivée des demandes. La brancarderie doit déterminer quel véhicule doit servir quelle demande en prenant en compte un objectif déjà fixé. Ce dernier peut-être la minimisation des retards, des distances parcourues ou des coûts. De façon générale, les demandes dans ce domaine sont imprévisibles et sont révélées de manière dynamique tout au long de la journée. Ce qui force le planificateur au sein de la brancarderie de prendre des décisions instantanées en temps réel. Ces décisions doivent prendre en considération les itinéraires actuels des brancardiers et la disponibilité des véhicules. La prise de décision dans ce cas est complexe puisqu'elle doit tenir compte des informations relatives aux demandes comme le délai de livraison de patient à respecter, le personnel médical disponible, le nombre de places nécessaires pour chaque patient et le niveau d'urgence de la demande.

Ce mémoire aborde le problème de transport interne dans les hôpitaux en considérant le cas du grand hôpital italien, le Policlinico Sant'Orsola-Malpighi. Cet hôpital compte plus de 30 pavillons dont chacun représente un service hospitalier distinct. Chaque année, il accueille un nombre énorme de patients qui souffrent de différentes maladies et dans des conditions différentes. En moyenne, ce dernier reçoit chaque jour plus de 425 patients, nécessitant des déplacements entre les différents pavillons afin de recevoir des soins particuliers.

Pour répondre aux demandes de transport des patients, l'hôpital dispose d'un service de transport (brancarderie) qui dispose d'une flotte de cinq véhicules, dont trois mini-véhicules électriques, et deux ambulances. Ce qui rend la tâche de répondre à ces demandes difficile, est que la majorité des demandes de transport (plus de 90%) arrivent de manière dynamique au cours de la journée, révélant différentes informations et exigences telles que le lieu de cueillette et de livraison et le nombre de places requises, le moment de début de service de transport et le délai de livraison à respecter. La durée des fenêtres de temps dépend du degré d'urgence de chaque demande. Plus le degré d'urgence est élevé, plus la fenêtre de temps dans laquelle les patients doivent être transportés sera restreinte.

Un tel contexte hospitalier crée le besoin ascendant pour développer un nouveau système d'aide à la décision capable de prendre des décisions en temps réel à l'arrivée des demandes de transport et capable d'optimiser les flux de transport sur le grand campus de l'hôpital.

Dans ce mémoire, nous développerons des outils d'aide à la décision pour les contextes hospitaliers et nous les appliquerons sur le cas réel de l'hôpital de Bologne. Nous avons considéré le retard total pondéré comme un critère d'évaluation de l'efficacité du système de planification de transport. Ce retard représente le temps dépassé dans les fenêtres de temps par les brancardiers et il est pondéré en fonction de l'urgence de chaque demande. L'objectif de ce travail est de minimiser le retard pondéré de toutes les requêtes de transport effectuées. Ce

qui permet de maximiser la qualité du service de transport offert et de minimiser les délais d'attente pour les patients. Aussi, ceci permet de réduire les coûts engendrés par l'occupation des ressources, comme les théâtres opératoires ou l'inactivité du personnel médical en attente des patients.

Nous abordons ce problème dans ces trois versions, statique, dynamique et stochastique en utilisant trois approches différentes. De plus, une plateforme de simulation du système de transport hospitalier réel a été développée pour tester les différentes approches et les différentes stratégies qui peuvent aider à optimiser notre fonction objectif.

Le premier chapitre de ce mémoire présente une revue de la littérature sur les problèmes de cueillettes et de livraisons et les problèmes de transport à la demande ainsi qu'une revue sur les problèmes des brancardiers. Ce chapitre offre également une revue sur les méthodes de résolution utilisées pour résoudre ces problèmes. Le deuxième chapitre est dédié à la présentation de notre problématique, du cadre général et de la modélisation du problème. Le troisième chapitre représente le cœur de ce travail et décrit les approches de résolution développées pour résoudre notre problème. Finalement, le quatrième chapitre est consacré à détailler les résultats obtenus par les différentes approches en faisant une simulation avec les données réelles d'une année entière.

Chapitre 1

Revue de la littérature

Introduction

Ce chapitre présente une synthèse des travaux et des études effectués en relation avec le problème de brancardiers. Dans un premier temps, nous allons commencer par présenter le problème de tournées de véhicules suivi d'une revue des problèmes de cueillettes et de livraisons. Ensuite, nous présenterons une revue sur les problèmes de brancardiers plus connus sous le nom de problèmes de transport de patients dans les hôpitaux. Cette revue contiendra également une présentation des différentes méthodes et approches de résolutions qui ont un lien avec notre travail. Finalement, nous mettrons tout ça en lien avec notre projet de recherche en présentant notre contribution à la recherche.

1.1 Les problèmes de tournées de véhicules

L'un des problèmes les plus connus dans la littérature et plus particulièrement dans le domaine de la recherche opérationnelle est le problème de voyageur de commerce (TSP). L'objectif de ce problème est de trouver un itinéraire qui minimise la distance totale parcourue pour un voyageur qui visite exactement une fois chaque ville d'un ensemble avant de revenir à son point de départ. Ce problème est NP-difficile et a été résolu pour la première fois par [Dantzig et al. \(1954\)](#).

Le problème de tournées de véhicules (VRP) est une généralisation du TSP. Introduit pour la première fois en 1959 par [Dantzig et Ramser \(1959\)](#), le VRP représente les problèmes réels des entreprises. Ce dernier se distingue du TSP par son utilisation d'une flotte complète de véhicules avec une capacité définie pour servir des clients ayant différentes demandes.

Après plus de 50 ans, le VRP compte aujourd'hui plusieurs variants. [Toth et Vigo \(2014\)](#) ont présenté une classification qui permet d'identifier les caractéristiques spécifiques d'un VRP à modéliser et à résoudre. Dans cette recherche, les problèmes ont été classés selon

la structure du réseau routier, le type de demande de transport, les contraintes qui affectent chaque itinéraire individuellement, la composition et la localisation de la flotte, les contraintes inter-routes et les objectifs d'optimisation.

Dans la littérature, nous trouvons une variante bien connue du VRP qui consiste à trouver des itinéraires entre deux lieux particuliers dont le premier est où quelqu'un ou quelque chose est récupéré, et le deuxième correspond au lieu de livraison. Dans un contexte de transport de marchandises, cette variante est appelée un problème de cueillettes et de livraisons (PDP). Cependant, dans un contexte du transport de personnes, le problème est connu sous le nom du problème de transport à la demande (DARP). Ces deux problèmes sont NP-difficiles.

Dans ce mémoire, le problème de brancardiers a été modélisé comme étant un problème de cueillettes et de livraisons. Ainsi, notre objectif est de trouver des itinéraires pour transporter des patients d'un service à un autre au sein d'un grand hôpital en Italie tout en respectant un ensemble de contraintes spécifiques.

1.2 Les problèmes de cueillettes et de livraisons

Une partie des problèmes de tournées de véhicules réside dans la famille des problèmes de cueillettes et de livraisons (PDP). Selon Battarra *et al.* (2014), nous pouvons trouver trois catégories des PDP qui se distinguent par le type de demande et la structure des itinéraires. Premièrement, nous trouvons les problèmes plusieurs-à-plusieurs qui concernent des produits ayant des origines et des destinations multiples où tout lieu peut être l'origine ou la destination de plusieurs produits. Deuxièmement, il existe les problèmes un-à-plusieurs-à-un qui sont caractérisés par des produits à livrer d'un dépôt aux clients et d'autres produits à collecter chez les clients et à ramener au dépôt. Finalement, les problèmes un-à-un où chaque produit a une seule origine et une seule destination entre lesquelles il doit être transporté. Notre problème s'inscrit dans cette dernière catégorie, puisqu'il consiste à transporter des patients entre deux localisations uniques à l'intérieur de l'hôpital.

L'incertitude et l'instabilité des conditions du système créent d'autres variants importants du PDP. En fait, si les informations pertinentes sur les conditions du système sont totalement ou partiellement dévoilées lors de son fonctionnement alors ce système peut être classé comme un problème dynamique. Si les conditions du système sont incertaines, nous pouvons dire que ce problème est stochastique et nous utilisons une distribution de probabilité pour décrire l'incertitude (Toth et Vigo, 2014).

Généralement, dans les problèmes dynamiques, les informations relatives à la localisation et à la demande des clients sont révélées au fil du temps. Cependant, certaines de ces informations sont connues à l'avance, mais pour les clients restants, ce n'est que des informations probabilistes qui sont données (Bektaş *et al.*, 2014). Dans les problèmes stochastiques, nous décrivons la demande des clients et les temps de déplacement comme des variables aléatoires puisqu'ils

sont incertains (Gendreau *et al.*, 1996). En conséquence, les itinéraires planifiés peuvent engendrer des retards de livraison pour les clients. En plus, ils peuvent être interrompus quand la capacité du véhicule est atteinte. Du coup, l'objectif principal des problèmes stochastiques est d'analyser l'impact de l'incertitude sur les coûts et les niveaux de service (Toth et Vigo, 2014).

En ce qui concerne le PDP dynamique un-à-un, un cadre général a été proposé dans l'étude de Berbeglia *et al.* (2010). Dans ce travail, deux types d'événements peuvent être trouvés. Le premier, lorsqu'un véhicule vient de terminer une demande et le deuxième, lorsque de nouvelles demandes arrivent. Dans le cas du premier événement, deux actions peuvent être prises, soit attendre soit se déplacer. Dans le cas du deuxième événement, les actions possibles sont soit d'accepter ou de rejeter les nouvelles demandes.

Pillac *et al.* (2013) ont présenté une revue sur les problèmes dynamiques en tenant compte du degré de dynamisme. Dans leur travail, nous trouvons une classification des problèmes de tournées de véhicules d'un point de vue de la qualité et de l'évolution de l'information. En fait, selon les auteurs, ces deux dimensions sont importantes et indispensables dans les problèmes logistiques réels. La qualité de l'information reflète l'incertitude des données. En d'autres termes, plus les données sont incertaines, plus la qualité de l'information décroît et vice-versa. Ces conditions ajoutent au problème la notion du dynamisme, ce qui impose la replanification des itinéraires actuels lors de leurs exécutions. L'évolution de l'information a un impact sur les tournées. En fait, les itinéraires peuvent être modifiés à cause de la disponibilité des nouvelles données, comme l'arrivée de nouvelles demandes de clients.

Dans la littérature, il existe des problèmes où nous trouvons des informations dynamiques et stochastiques en même temps. Ritzinger *et al.* (2016) ont présenté une revue sur ces problèmes qui combinent des informations dynamiques et stochastiques. Dans cet article, les auteurs ont examiné des travaux purement dynamiques, d'autres purement stochastiques et la combinaison des deux. De plus, ils ont présenté une classification basée sur les algorithmes d'optimisation.

Dernièrement, un problème qui est généraliste du PDP un-à-un attire l'attention des chercheurs. Ce dernier est le problème de livraison le jour même (SDDP) qui a été introduit et traité par Voccia *et al.* (2019). Ce problème est considéré comme un PDP dynamique avec des fenêtres de temps rigides et des véhicules identiques basés dans un dépôt central où les marchandises sont disponibles pour la cueillette. Les auteurs ont employé une fonction de consensus basée sur des scénarios échantillonnés pour prendre des décisions dynamiques et, par conséquent, modifier les itinéraires actuels. Ils ont également utilisé une stratégie d'attente pour permettre à un véhicule d'attendre au dépôt alors que de nouvelles demandes arrivent. L'objectif est de minimiser le nombre de demandes non servies. Pour ce faire, les auteurs ont utilisé des prédictions basées sur des demandes déjà connues et des demandes fictives échan-

tillonnées des distributions de probabilité connues. Les résultats expérimentaux ont montré que la prise en compte de l'incertitude des demandes fictives a un impact significatif sur la qualité de la solution.

Bien que le SDDP et le problème traité dans ce mémoire semblent un peu différents en termes d'application, en réalité, ils partagent beaucoup de caractéristiques ensemble. En fait, dans les deux problèmes, les véhicules effectuent plusieurs trajets courts avec une capacité limitée et de courtes fenêtres de temps. De plus, les deux problèmes partagent une autre caractéristique pertinente qui est la date de disponibilité des demandes. Cette date représente le moment où les marchandises ou les patients, dans notre cas, deviennent disponibles pour la cueillette. Du coup, aucune demande ne peut être intégrée dans une route qu'après cette date.

Dans le même contexte, les travaux de recherche de Côté *et al.* (2021) ont également abordé le problème de la livraison le jour même (SDDP). Ces travaux ont traité le problème de tournées de véhicules dynamique où des clients stochastiques demandent des livraisons urgentes caractérisées par des fenêtres de temps restreintes. L'objectif est de maximiser le nombre de demandes servies tout en minimisant la distance parcourue. Les auteurs ont proposé une nouvelle approche pour modéliser ce problème comme un PDP dynamique avec des fenêtres de temps et des dates de disponibilité pour les demandes. L'approche de résolution repose sur des algorithmes dynamiques, tels que l'heuristique de réoptimisation, et d'autres stochastiques comme l'approche de planification basée sur des scénarios échantillonnés et l'heuristique de séparation et de regret (B&R). Les scénarios échantillonnés utilisés dans les heuristiques stochastiques sont générés à chaque fois qu'un nouvel événement se produit. Ensuite, ils sont optimisés afin d'obtenir un plan réalisable qui peut être mis en œuvre dans tous les scénarios et qui conduit à des solutions à faible coût. Les auteurs ont développé deux nouvelles fonctions de consensus dans le but de sélectionner les plans d'itinéraires à mettre en œuvre.

1.3 Les problèmes de transport à la demande

Le DARP se distingue des autres problèmes de cueillettes et de livraisons en se concentrant sur la maximisation de la satisfaction des utilisateurs et sur la minimisation des coûts de transport. La maximisation de la satisfaction des clients se fait à travers la réduction du temps nécessaire pour un service. En fait, ce qui distingue plus le DARP du problème de cueillettes et de livraisons est la contrainte du trajet maximal qui garantit que les passagers ne passeront pas plus qu'un temps maximal à l'intérieur des véhicules. Ces caractéristiques engendrent plus des coûts d'investissement et d'exploitation des véhicules. Généralement, ces coûts sont liés au nombre de véhicules utilisés, la distance parcourue et la durée totale du parcours. En conséquence, l'objectif du DARP est de construire un ensemble d'itinéraires capable d'optimiser et d'équilibrer ces objectifs contradictoires. Dans les cas réels, les itinéraires doivent être conçus dans une très courte durée. Ce qui explique la domination de l'utilisation

des heuristiques et des métaheuristiques dans la littérature pour traiter des problèmes de grande taille.

Dans le domaine de transport des personnes, Madsen *et al.* (1995) ont résolu un DARP dynamique multi-objectif traitant du transport de personnes âgées et handicapées. Le problème contient des contraintes de capacité, d'une durée maximale pour le trajet, d'écart maximal entre le temps de parcours réel et le temps de parcours le plus court possible, de fenêtres de temps, d'une flotte hétérogène et d'urgence des patients. Les auteurs ont utilisé une heuristique d'insertion pour mettre à jour les itinéraires actuels. L'algorithme développé a été testé avec des données réelles du service de pompiers de Copenhague. Dans le but de mesurer la qualité des solutions, plusieurs indices de performance liés à la satisfaction et aux dépenses par patient ont été utilisés. Les résultats ont montré que l'algorithme peut être appliqué en ligne dans un environnement dynamique.

Doerner et Salazar-González (2014) considèrent que le DARP est l'un des problèmes les plus connus dans le domaine du transport de personnes. En fait, ce dernier consiste à concevoir des itinéraires pour des passagers ayant un lieu spécifique pour la cueillette et un lieu spécifique pour la livraison. Les auteurs ont défini le DARP comme une généralisation du problème de tournées de véhicule de cueillettes et de livraisons (PDP) avec des fenêtres de temps, en plus d'intégrer des contraintes spécifiques aux aspects humains qui garantissent un service de haute qualité.

Ho *et al.* (2018) ont présenté une revue bien détaillée sur le DARP dans laquelle ils ont considéré 86 articles depuis 2007. Les auteurs ont présenté quelques domaines d'application du DARP, les tendances et les défis futurs ainsi qu'une classification des méthodes de résolution et des variantes du problème. Cette classification est basée sur quand une décision est prise, c'est-à-dire statique ou dynamique, et sur la certitude de l'information lors de la prise de décision, en d'autres termes, déterministe ou stochastique.

Luo *et al.* (2019) ont considéré une autre variante du DARP où il y'a plusieurs trajets, et des contraintes telles que des véhicules hétérogènes, des fenêtres de temps, des multiples types de demandes, une durée de trajet maximale, une capacité de véhicule configurable et une planification de la main-d'œuvre. L'objectif est d'abord de maximiser le nombre de demandes satisfaites, ensuite, de minimiser la distance totale parcourue par les véhicules. Pour résoudre ce problème, ces auteurs ont proposé un algorithme exact à deux phases de "branch and price and cut". Cet algorithme exact en deux phases a été testé et les résultats ont montré que ce dernier résout plusieurs instances de manière optimale et surpasse d'autres méthodes proposées dans la littérature. La plus grande instance testée est composée de 37 demandes. Cependant, sur les 41 instances de test, l'algorithme résout 27 instances à l'optimalité en quatre heures dont la plus grande instance est composée de 36 demandes.

En ce qui concerne le problème de brancardiers traité dans ce mémoire, nous ne prenons pas

en compte le temps passé par les patients à l'intérieur des véhicules, une contrainte commune dans le DARP, puisque les fenêtres de temps sont courtes et que l'objectif est de minimiser le retard. Pour cette raison, notre problème est considéré comme un PDP.

Une description détaillée du problème étudié sera donnée dans le chapitre 2.

1.4 Le problème des brancardiers

Une autre application intéressante du DARP se trouve dans le domaine de la santé. Ce domaine se caractérise par l'urgence des demandes de transport et par les fenêtres de temps de courte durée. En plus, plusieurs autres facteurs doivent être pris en considération, tels que la compatibilité de l'équipement et du personnel. En outre, le problème d'ordonnancement du personnel et de la maintenance augmente également la complexité du DARP dans le domaine de la santé.

Odegaard *et al.* (2007a) ont étudié, pendant sept mois, les opérations des brancardiers dans un hôpital à Vancouver. La recherche a présenté les différentes opérations du système pendant la durée d'étude, les défis rencontrés et les mesures de performances développées ce qui a souligné l'importance d'avoir un service de brancardiers efficace. Ce qui joint la recherche effectuée par Odegaard *et al.* (2007b) qui a proposé un modèle de simulation pour mesurer l'impact des changements de système dans un hôpital réel ainsi qu'un programme linéaire pour optimiser l'horaire du personnel.

Melachrinoudis *et al.* (2007) ont proposé un DARP avec des fenêtres de temps flexibles appliqué dans un centre pour les services de santé et de rétablissement du comportement addictif à Boston. L'objectif est de minimiser une combinaison du coût total de transport des patients par les véhicules et du temps total de dérangement des patients. Ce dernier représente le temps total passé dans un trajet excessif, un temps de livraison précoce ou tardif avant le service et un temps de ramassage tardif après le service. Le problème a été modélisé en faisant recours à la programmation en nombres entiers mixtes et résolu à l'aide d'une heuristique de recherche tabou. Toutefois, le problème a été traité dans sa version statique et déterministe où toutes les demandes de transport sont connues d'avance et les événements incertains ne sont pas pris en considération.

Fiegl et Pontow (2009) ont développé un algorithme en ligne afin de planifier les tâches de cueillettes et de livraisons dans les hôpitaux. Dans ce problème, les tâches de transport arrivent au fil du temps, et donc, le problème est considéré dynamique. L'approche de résolution utilise des techniques de la théorie classique de l'ordonnancement et de la théorie des graphes. L'objectif est de minimiser le temps de flux moyen pondéré, et du coup, le temps d'attente et les coûts des patients. L'algorithme a été utilisé à l'hôpital national de Natters en Autriche et les résultats ont montré qu'une telle approche à une performance est supérieure à celle du système actuel de planification.

Pour faire face à un DARP dynamique avec des contraintes spécifiques à l'hôpital, Hanne *et al.* (2009) ont conçu un système de planification informatisé qui prend en charge toutes les phases du flux de transport au sein des hôpitaux, y compris la réservation des requêtes de transport, l'affectation des demandes de transport et la surveillance ainsi que le suivi des trajets en temps réel. Quatre critères de performance pour évaluer la qualité et le coût de la solution du problème ainsi que quatre sources d'incertitude pour le problème de transport des patients ont été défini avant de traiter ce problème. Les sources sont : la majorité des requêtes de transport sont non connus à l'avance, les exigences des demandes précédemment réservées peuvent changer, les dates d'arrivée et de départ exactes associées à chaque localisation ne sont pas connues à l'avance et enfin l'absence d'un membre de l'équipe des brancardiers. Ils ont proposé plusieurs algorithmes d'optimisation nécessitant des temps de résolution différents. Ce système de planification a contribué à améliorer la satisfaction des patients et à réduire les coûts de transport de 20%.

Dans le même contexte, Beaudry *et al.* (2010) ont proposé une analyse et une méthode de résolution du problème de transport des patients survenant dans les grands hôpitaux. L'objectif est de fournir un service de transport efficace et ponctuel pour les patients entre les différents services dans un campus hospitalier. C'est un problème multi-objectif, visant à minimiser une somme pondérée de temps de trajet total, de retard total et de précocité totale. Il contient des contraintes liées aux fenêtres de temps, une flotte hétérogène, des pannes de véhicules, des pauses de repos inattendues et des dépôts multiples. Différemment de notre travail, ils n'ont permis que des trajets simples. Dans un tel environnement, les demandes de transport arrivent de manière dynamique ce qui demande une solution capable d'insérer rapidement de nouvelles demandes dans les itinéraires actuels du véhicule. Pour ce faire, les auteurs ont proposé une procédure heuristique en deux phases où une solution réalisable est générée dans la première phase grâce à une heuristique d'insertion. Ensuite cette solution est améliorée dans la deuxième phase en utilisant un algorithme de recherche tabou. En plus, ils ont mené des expériences sur des données réelles fournies par un hôpital allemand. Les résultats ont montré que l'algorithme est capable de manipuler l'aspect dynamique du problème. Encore, nous pouvons obtenir des solutions de haute qualité à l'aide de la recherche tabou tout en réduisant le temps d'attente des patients et le nombre de véhicules utilisés.

Kergosien *et al.* (2011) ont étudié aussi le problème de transport de patients entre les unités de soins dans un complexe hospitalier de Tours en France. Le problème a été modélisé comme un PDP dynamique où chaque demande de transport nécessite un type de véhicule spécifique et un véhicule ne peut transporter qu'une seule personne à la fois. La sous-traitance des demandes de transport est possible, mais engendre des coûts élevés. La planification du transport des patients doit respecter des contraintes particulières telles que, la priorité des demandes urgentes, la désinfection d'un véhicule après le transport d'un patient atteint d'une maladie contagieuse et le respect du type de véhicule nécessaire. L'objectif est de minimiser

la somme des coûts de transport pour l'hôpital et les entreprises privées. Pour ce faire, les auteurs ont développé un algorithme dynamique et une procédure de recherche tabou. Des expériences avec des instances réelles et des instances générées aléatoirement ont montré que cette approche peut fournir des solutions de haute qualité pour ce problème dynamique avec un temps de calcul court, de moins de 5 secondes en moyenne.

Schmid et Doerner (2014) ont traité le problème d'ordonnancement des examens et des salles d'opération ainsi que l'optimisation simultanée du transport des patients dans un hôpital. Les chercheurs ont présenté un modèle d'optimisation combinatoire intégratif combinant les aspects liés à l'ordonnancement et au VRP. L'objectif est de minimiser le temps d'attente des patients, le temps mort dans les salles, et les coûts liés aux déplacements vides des brancardiers. En général, les deux sous-problèmes essentiels sont résolus indépendamment, mais les auteurs ont utilisé une métaheuristique hybride coopérative qui présente des avantages substantiels par rapport aux processus d'optimisation hiérarchique découplés. Les résultats expérimentaux ont montré qu'on peut obtenir des solutions de haute qualité en un temps d'exécution raisonnable. Pour les petites instances, la solution optimale a été trouvée en quelques secondes. Néanmoins, la performance de la métaheuristique proposée a été testée en utilisant uniquement une version statique et déterministe du problème.

Zhang *et al.* (2015) ont travaillé sur un problème réel du transport des patients avec l'autorité hospitalière de Hong Kong. Ce problème a été modélisé comme un problème de transport à la demande (DARP) multi-trajets, nécessitant la conception de plusieurs itinéraires pour chaque ambulance. Chaque itinéraire conçu, doit respecter différentes contraintes tel que les contraintes de précédences, la contrainte de capacité, le temps de trajet et les fenêtres de temps. L'intérieur des ambulances doit être désinfecté au dépôt entre deux trajets consécutifs. L'objectif principal est de répondre au plus grand nombre de demandes de transport possible avec les ressources données. Pour ce faire, les auteurs ont proposé une formulation mathématique du problème et ont développé un algorithme avec un opérateur de recombinaison personnalisé afin de le résoudre. Dans cet article, la méthode d'insertion et de regret est utilisée pour construire la population, et la technique d'évaluation basée sur les segments est adaptée pour évaluer les solutions temporelles. Une évaluation de l'algorithme a été faite grâce à des données du monde réel de 2009. Pour les petites instances, une comparaison avec les résultats obtenus en résolvant le modèle mathématique a démontré que l'algorithme est capable de les résoudre rapidement jusqu'à l'optimalité.

Dans le travail de Detti *et al.* (2017), un DARP multi dépôt issu d'une application réelle dans le domaine de la santé, concernant le transport non urgent des patients a été présenté et résolu. Ce problème doit respecter plusieurs contraintes, tel que l'utilisation d'une flotte de véhicules hétérogènes, la compatibilité des véhicules-patient, la qualité de service, les préférences des patients et les coûts liés à l'attente des véhicules. Le problème a été modélisé à l'aide d'une formulation de programmation linéaire mixte en nombres entiers. Pour respecter toutes ces

contraintes, les auteurs ont proposé des algorithmes de recherche de voisinage variable et de recherche tabou. Les résultats des expériences menées sur de grandes instances réelles et aléatoires montrent l'efficacité des approches proposées.

Moons *et al.* (2019) ont évalué la chaîne logistique interne de l'hôpital et plus particulièrement dans la salle d'opération, puisqu'elle fait partie des ressources les plus critiques d'un hôpital. Dans cette dernière, les items demandés devraient être disponibles au bon moment, au bon endroit, dans le bon état et au plus bas coût possible. Les auteurs ont étudié les articles publiés sur ce sujet et ils ont discuté des mesures de performances basées sur des approches et des techniques de prise de décision multicritères. Le résultat de cette recherche a prouvé la nécessité d'une étude de cas dans la salle d'opération serait nécessaire afin d'illustrer et valider les mesures de performance étudiées.

1.5 Méthodes de résolution

Au cours des dernières années, plusieurs approches de résolution ont été proposées dans le but de résoudre le problème de transport de patients dans les hôpitaux. Généralement, nous pouvons classer les différentes méthodes utilisées pour résoudre ce problème en deux grandes catégories. La première est celle des méthodes exactes. Ces méthodes visent à trouver une solution optimale pour le problème donné à travers une recherche complète de l'espace de toutes les combinaisons possibles. La deuxième catégorie est celle des méthodes approximatives. Ces dernières permettent de trouver une bonne solution, en un temps raisonnable, mais ne garantissent pas l'optimalité. En pratique, dans les problèmes où l'on a plusieurs demandes de transport, au-delà de quelques dizaines, les méthodes exactes nécessitent un temps énorme d'exécution et ne sont pas capables de trouver la solution optimale. Tandis que dans les cas réels, comme notre problème, il est primordial que les temps d'exécution des algorithmes soient très rapides, à savoir instantanés, pour garantir une planification dynamique en temps réel.

Dans cette section, nous allons présenter des approches et des techniques de résolution présentées dans la littérature qui ont inspiré notre travail et ont un lien avec notre problème et notre méthodologie de résolution.

1.5.1 Méthodes de résolution statiques

Dans la littérature, différentes méthodes exactes pour résoudre le problème de cueillettes et de livraisons statique ont été présentées. Le premier travail qui a présenté une méthode exacte pour résoudre le PDP statique avec un seul et plusieurs véhicules et celui de Kalantari *et al.* (1985). Ce travail traite le PDP sans et avec des contraintes de capacité. Les auteurs ont utilisé une extension de l'algorithme de séparation et évaluation (B&B) de Little *et al.* (1963). Les résultats pour le cas d'un seul véhicule et avec des contraintes de capacité ont montré que

l'algorithme est capable de trouver des solutions dans un temps de calcul raisonnable pour des instances jusqu'à 31 clients.

Ropke et Cordeau (2009) ont présenté un nouvel algorithme de branch-and-cut-and-price pour le PDP avec des fenêtres de temps. Ce dernier utilise la génération de colonne pour résoudre la relaxation du programme linéaire, ce qui permet de calculer les bornes inférieures. Pour les sous-problèmes, les auteurs ont résolu un problème de plus court chemin élémentaire et un autre non élémentaire. Aussi, les auteurs ont ajouté des inégalités valides. En se basant sur les résultats trouvés, et pour les instances tests considérées, les auteurs ont recommandé l'utilisation du problème du plus court chemin élémentaire, comme sous-problème car il donne les meilleurs résultats.

En ce qui concerne les heuristiques pour le PDP statique, l'un des premiers travaux proposés est celui de Van der Bruggen *et al.* (1993). Les auteurs ont considéré un PDP avec des fenêtres de temps, un seul dépôt, un seul véhicule et des contraintes de capacité. Pour résoudre ce problème, ils ont développé une méthode de recherche locale sur deux étapes basées sur une recherche à profondeur variable. Dans la première étape, une solution réalisable est créée. Ensuite, dans la deuxième étape, la solution est améliorée. La méthode a été appliquée sur un cas réel et a été capable de trouver des solutions quasis optimales dans un temps d'exécution acceptable. De plus, pour garantir que la méthode trouve toujours des solutions réalisables et de bonnes qualités, ils ont proposé un autre algorithme basé sur le recuit simulé. Ce dernier peut garantir des solutions de haute qualité, mais en contrepartie, il demande un temps d'exécution important.

Bent et Van Hentenryck (2006) ont résolu le problème de cueillettes et de livraisons avec des fenêtres de temps et des multivéhicules avec un algorithme hybride en deux étapes. Dans la première étape, un algorithme de recuit simulé est utilisé afin de réduire le nombre d'itinéraires. Dans la deuxième étape, une de recherche à grand voisinage est employée pour diminuer le coût total du trajet. L'algorithme hybride a pu trouver de nouvelles meilleures solutions pour des instances connues dans la littérature de 200, 600 et 1000 clients. Plus particulièrement, cette méthode a amélioré les instances de 200 clients de 47% et celles de 600 clients de 76%.

Ropke et Pisinger (2006) ont résolu le problème de cueillettes et livraisons avec des fenêtres de temps en proposant une heuristique performante basée sur une extension de l'heuristique de recherche à grand voisinage qui est la recherche adaptative à grand voisinage (ALNS). La procédure suppose qu'une solution initiale a été déjà trouvée. Pour ce faire, il est possible d'utiliser une heuristique de construction par exemple. L'ALNS utilise un certain nombre de sous-heuristiques concurrentes avec une fréquence qui dépend de leur performance historique. Sur 350 instances de benchmark, allant jusqu'à 500 demandes, cet algorithme était capable, pour 50% des cas, d'améliorer les meilleures solutions connues de la littérature et pour 80%

des cas il était capable de trouver une solution qui est au moins aussi bonne que la solution précédente la plus connue.

1.5.2 Méthodes de résolution dynamiques et stochastiques

Dans la littérature de la recherche opérationnelle, les problèmes de tournées de véhicules dynamiques et stochastiques ont une histoire de 30 ans. Cependant, selon Ulmer *et al.* (2020), l'intérêt des chercheurs sur ses problèmes n'a augmenté que récemment. Ils ont expliqué cette tendance par la disponibilité des données et la puissance de calcul qui augmente de jour en jour.

Powell (2007) a présenté quatre politiques principales pour classer les différentes stratégies algorithmiques utilisées dans les problèmes dynamiques et stochastiques pour prendre les décisions. Ces politiques sont : les politiques de myopes, les politiques d'anticipation, les approximations de fonctions de politique et les approximations de fonctions de valeur. Ensuite, Ulmer *et al.* (2020) ont utilisé ces politiques pour définir quatre types de méthodologie afin de classer les différentes méthodes de résolution des problèmes de tournées de véhicules dynamiques et stochastiques. Dans cette étude, les auteurs ont défini un plan d'itinéraires comme une solution partielle pour le problème dynamique. Les quatre types de méthodologie sont :

- La réoptimisation : le plan d'itinéraires est établi en se basant sur les informations actuellement disponibles pour le planificateur.
- Les algorithmes d'anticipation : ces algorithmes utilisent des informations stochastiques afin de construire des plans d'itinéraire.
- Les approximations de fonctions de politique : le plan d'itinéraires est établi à partir d'une technique qui imite la prise de décision dans la pratique.
- Les approximations de fonctions de valeur : cette catégorie utilise des techniques de simulation et d'entraînement répété pour apprendre la valeur d'une décision.

Le reste de cette section porte sur les deux premières méthodologies.

Dans les premiers travaux de recherche sur les problèmes de tournées dynamiques et stochastiques, les méthodes de réoptimisation ont été la base de la prise de décision. Ces méthodes utilisent un horizon roulant afin de sélectionner, à chaque époque, un plan d'itinéraires pour un problème de tournées statique qui contient des informations connues et qui ne considère pas l'aspect stochastique.

Parmi ces travaux, nous pouvons citer Gendreau *et al.* (1999) qui ont étudié le problème de tournées de véhicules dynamiques avec des fenêtres de temps souples inspirés par un cas de service de messagerie. Dans ce problème, les demandes des clients doivent être affectées en temps réel à une flotte de véhicules déjà en service. Pour ce faire, les auteurs ont commencé

par développer une heuristique de recherche tabou pour résoudre la version statique de ce problème. Ensuite, cette heuristique a été adaptée pour résoudre le cas dynamique. Dans le but d'augmenter l'effort de calcul, l'implémentation de cette dernière a été faite sur une plateforme parallèle. Les auteurs ont comparé l'heuristique de recherche tabou avec d'autres méthodes heuristiques et les résultats ont montré que leur méthode permet de servir plus de clients et de réduire la distance totale parcourue ainsi que le retard total.

Bien que les méthodes de réoptimisation ne prennent pas en compte les prédictions des demandes futures, elles sont souvent utilisées dans la littérature récente. À titre d'exemple, Schyns (2015) a traité un problème de camions de ravitaillement en carburant dans un aéroport. Ce problème est dynamique avec des livraisons fractionnées et des fenêtres de temps qui changent d'une manière stochastique tout au long de l'horizon de planification. De plus, il existe des contraintes de capacité et de flotte de véhicules hétérogènes. Pour résoudre ce problème, les auteurs ont développé un nouvel algorithme basé sur un système de colonies de fourmis. L'objectif de ce dernier est de minimiser la distance et d'optimiser la réactivité (le temps de réponse). L'algorithme a été testé sur un ensemble d'instances réelles et artificielles, et les résultats ont prouvé que ce dernier a été capable de garantir de bonnes solutions dans un intervalle de temps limité.

Selon Ulmer *et al.* (2020) les méthodes de réoptimisation sont courantes surtout dans les problèmes de tournées où il y a un nombre de contraintes opérationnelles important.

Avec l'avancement et le développement technologique et en particulier l'augmentation de la performance et de la puissance de calcul des ordinateurs, l'utilisation des algorithmes d'anticipation a augmenté. Ces derniers sont des algorithmes dynamiques qui prennent en considération l'aspect stochastique des problèmes. Généralement, ils utilisent des scénarios fictifs pour créer des plans d'itinéraires.

L'un de ces algorithmes les plus connus dans la littérature est l'approche de planification à scénarios multiples (MSA) présenté par Bent et Van Hentenryck (2004). Ce travail aborde les problèmes de tournées de véhicules partiellement dynamiques avec des clients stochastiques. En fait, certains clients sont connus au moment de la planification tandis que d'autres sont révélés au cours du temps et l'objectif est de maximiser le nombre de clients desservis. L'idée de l'approche consiste à générer continuellement des plans de tournées pour des scénarios comprenant des demandes connues et des demandes stochastiques des clients. Ensuite, un plan distingué sera sélectionné à l'aide d'une fonction de consensus qui choisit le plan le plus similaire aux itinéraires actuels. En comparant les résultats de cette approche avec celles qui n'exploitent pas l'information stochastique, les auteurs ont trouvé des améliorations importantes et ont montré qu'il y a une synergie entre les informations stochastiques et la fonction de consensus ce qui a amélioré la qualité des solutions de manière significative.

Un autre type d'algorithme d'anticipation a été présenté par Hvattum *et al.* (2006) qui ont

considéré un problème de tournées de véhicules où la localisation et la demande des clients peuvent être inconnues à l'avance. Le problème a été modélisé comme un SDVRP et il a été résolu à l'aide d'une heuristique de couverture dynamique et stochastique (DSHH) qui utilise des scénarios échantillonnés. Cette méthode permet de générer des scénarios échantillonnés, pour chaque intervalle de temps, et de les résoudre afin de construire un plan pour les tournées. Les auteurs ont comparé leur méthode avec l'heuristique de réoptimisation qui ne prend pas en compte les informations stochastiques. Ils ont trouvé que le DSHH a réussi à réduire la distance totale parcourue de 15%. Toutefois, ils ont remarqué que le nombre de véhicules utilisés a été augmenté avec cette méthode.

Afin de corriger cet inconvénient, *Hvattum et al. (2007)* ont présenté une heuristique de séparation et regret (B&R) pour traiter le VRP stochastique et dynamique. Ce travail est inspiré par un problème réel d'un grand transporteur en Norvège. Cette approche a été comparée avec l'heuristique de réoptimisation et a réussi à trouver de meilleurs résultats. Cependant, la distance parcourue était supérieure à celle du DSHH. Aussi, une comparaison avec l'approche MSA a été faite. Selon les résultats, la MSA est meilleure que la B&R en termes de nombre de clients non desservi qui a diminué d'une moyenne de 1.9 avec la B&R à 1.1 client avec la MSA. Les auteurs ont conclu que cette heuristique, qui intègre des informations stochastiques pendant le processus de résolution, reste supérieure aux heuristiques basées sur une approche purement dynamique. En plus, l'heuristique proposée est bien adaptée aux problèmes contenant des clients stochastiques, des demandes stochastiques ou les deux.

Une autre technique souvent utilisée avec les algorithmes d'anticipation est appelée la fonction de consensus. En fait, *Bent et Van Hentenryck (2004)*, *Voccia et al. (2019)*, *Song et al. (2020)* et *Côté et al. (2021)* ont tous utilisé des fonctions de consensus pour attribuer un score pour chaque solution en fonction du nombre de fonctionnalités présentes dans les autres solutions. L'objectif est de choisir la solution qui a la meilleure probabilité d'être appliquée à moindre coût dans tous les scénarios. *Voccia et al. (2019)* ont utilisé une fonction de consensus qui compte le nombre de fois où les itinéraires d'un plan apparaissent dans d'autres plans. *Côté et al. (2021)* ont utilisé deux fonctions de consensus. La première est appelée la similitude des affectations. Pour chaque couple demande-numéro d'itinéraire d'un plan, elle calcule un score qui représente le nombre de fois que ce couple apparaît dans d'autres plans. Le plan ayant le score le plus grand est sélectionné. La deuxième est appelée la modification de la distance qui calcule un score pour chaque plan qui représente la somme du nombre de changements nécessaires pour obtenir chacun des autres plans. Dans cette deuxième fonction de consensus, le plan qui a le score minimum est sélectionné.

Dans la littérature, il existe d'autres techniques pour gérer le dynamisme dans les problèmes de tournées de véhicules. Ces techniques sont les stratégies d'attente et les stratégies de relocalisation. Une stratégie d'attente indique à un véhicule d'attendre à l'endroit où il se trouve pour un certain temps au lieu de se déplacer vers son prochain client connu. Une

stratégie de relocalisation déplace des véhicules vers une localisation où une demande pourrait survenir. La possibilité d'attendre et de relocaliser permet d'anticiper les demandes futures potentielles pour réduire les retards. Pour chacune de ces stratégies, il existe deux types :

- Des stratégies implicites : celle-ci permettent à un véhicule d'attendre ou de se déplacer en utilisant une règle empirique pour déterminer l'endroit et le temps d'attente.
- Des stratégies explicites : sont les stratégies qui permettent à un véhicule d'attendre ou de se déplacer et elles utilisent des informations sur les demandes fictives pour déterminer où et combien de temps à attendre.

Mitrović-Minić et Laporte (2004) ont développé et comparé quatre stratégies d'attente pour faire face à un PDP dynamique découlant des compagnies de messagerie. Ces stratégies d'attente sont basées sur deux types de décisions, soit "conduire d'abord", soit "attendre d'abord". Les auteurs montrent que la distribution des temps d'attentes le long des itinéraires a un impact sur la qualité de la solution. En plus, nous pouvons anticiper les demandes futures en incluant des temps d'attentes dans les itinéraires.

Bent et Van Hentenryck (2007) ont résolu un problème de SDVRP en ligne avec multi-véhicules et des fenêtres de temps. L'objectif est de maximiser le nombre de clients desservis. Pour ce faire, les auteurs ont développé un algorithme qui utilise des stratégies d'attente et de relocalisation. L'algorithme résout des scénarios échantillonnés, et selon la meilleure évaluation trouvée, il décide automatiquement quand un véhicule doit attendre et où un véhicule doit se relocaliser. Les résultats ont prouvé l'efficacité de ces stratégies et ils ont montré que leur implantation améliore le nombre de clients desservis.

Vonolfen et Affenzeller (2016) ont examiné les méthodes de distribution de temps d'attente pour le PDP dynamique proposées dans la littérature. Ensuite, les auteurs ont proposé deux améliorations. La première consiste en une heuristique qui prend en considération des informations historiques ainsi que les points forts des stratégies d'attentes proposées dans la littérature. Tandis que la deuxième est une approche qui permet de paramétrer automatiquement les stratégies d'attente selon les caractéristiques du problème. Les auteurs ont trouvé que le réglage des paramètres des stratégies d'attente peut se faire automatiquement en utilisant des instances d'entraînement. De plus, les résultats ont montré que l'utilisation des données historiques avec des instances aléatoires et des fenêtres de temps restreintes n'est pas bénéfique et que la deuxième approche donne de meilleurs résultats.

En bref, les techniques et les méthodes de résolution citées dans cette section ont contribué à nous inspirer pour développer notre propre approche qui utilise une combinaison d'heuristiques efficaces, une approche de planification basée sur des scénarios, une fonction de consensus et des stratégies d'attente et de relocalisation développées spécifiquement pour notre problème. Plus de détails sur notre méthodologie de résolution seront présentés dans le

1.6 Contribution à la recherche

Ces dernières années, le problème de brancardiers ou de transport de patients dans les hôpitaux attire de plus en plus l'attention des chercheurs. Cet intérêt ascendant peut s'expliquer par plusieurs facteurs, tels que : le développement rapide de la technologie, l'exigence des clients, la mise en priorité de la satisfaction des patients et surtout des coûts énormes liés à la logistique interne des hôpitaux.

Dans les grands hôpitaux, le transport de patients entre les différents pavillons se fait soit avec des ambulances soit avec d'autres types de véhicules. Les demandes de transport sont dynamiques et donc dévoilées en temps réel. Ceci augmente la complexité et la difficulté de la planification et de la gestion des flux de transport de patients pour le service de brancarderie. De plus, ceci augmente de façon importante les coûts de transport pour l'hôpital. Dans cet environnement, plusieurs auteurs ont proposé différentes approches de résolution pour résoudre ce problème visant à augmenter la satisfaction des patients en premier lieu et à diminuer les coûts de transport en second lieu. Ces approches varient entre des programmes linéaires, des modèles de simulation, des algorithmes simples et dynamiques, des différentes heuristiques et des stratégies d'attentes et de relocalisations.

Le problème étudié dans ce mémoire pose plusieurs questions que le décideur, qui est le service de brancarderie dans notre cas, devrait y répondre lors de la construction des itinéraires pour les brancardiers. Parmi ces questions nous pouvons citer : quel type de véhicule à utiliser ? Quelle est la façon la plus efficace pour affecter les demandes de transports aux brancardiers et pour construire les itinéraires ? Après avoir terminé le service d'une demande de transport, est-ce qu'il sera mieux que le brancardier attend sur place une autre demande, ou le relocaliser vers une autre localisation ? Est-ce qu'on peut tirer des avantages et profiter des historiques des demandes de transport de l'hôpital pour nous aider à prendre des décisions en temps réel ? Dans ce contexte, nous proposons trois approches de résolution et trois stratégies qui peuvent aider les brancardiers à répondre à ces questions ci-dessus. En effet, nous avons résolu le problème de transport de patients dans un grand hôpital en Italie. Notre objectif est de minimiser le retard pondéré total et de minimiser les coûts de transport.

Plus spécifiquement, notre contribution consiste en :

- Une heuristique de réoptimisation dynamique adaptée pour notre cas qui utilise une heuristique de regret, de recherche locale et une recherche au grand voisinage adaptatif (ALNS).
- Trois stratégies de relocalisation adaptées pour notre problème.
- Une approche de planification basée sur des scénarios (SBPA) qui prend en considération des demandes fictives échantillonnées par une distribution de probabilité.

- Une fonction de consensus qui utilise des informations stochastiques afin de choisir une relocalisation pour un brancardier après avoir considéré tous les scénarios.

Même si ces algorithmes sont développés pour résoudre un cas spécifique, nous croyons qu'ils peuvent résoudre également les autres problèmes similaires.

Chapitre 2

Le problème des brancardiers

Introduction

Ce chapitre est dédié à la présentation du contexte général de ce travail. Nous commençons par définir la problématique traitée dans ce mémoire. Ensuite, nous présenterons les hypothèses de simplification considérées. Cela sera suivi par une description des processus de planification suivie actuellement par le service de brancarderie de notre cas et une analyse des données réelles collectées de l'hôpital. Finalement, une modélisation du problème sera donnée.

2.1 Problématique

Avec plus de 400 ans d'histoire, l'hôpital universitaire Policlinico Sant'Orsola-Malpighi est le plus grand hôpital de la mairie de Bologne. Il est classé comme étant le plus grand hôpital en Italie en termes de nombre de lits avec plus de 1 758 lits, et environ 5 355 employés, dont 881 médecins. Chaque année, il reçoit environ 72 000 cas d'hospitalisation et environ 4 000 000 de visites ambulatoires par des médecins spécialistes. Il accueille en moyenne 20 000 personnes par jour, dont les patients et le personnel. En termes de superficie, le campus de cet hôpital s'étend sur plus de 1.8 km de long avec 31 pavillons autour d'une avenue centrale bordée d'arbres, comme illustré dans la Figure 2.1.

Cet hôpital offre un service de transport pour ses patients qui doivent se déplacer entre les pavillons en utilisant une flotte de cinq véhicules, dont trois véhicules électriques et deux ambulances, conduits par des brancardiers tout au long de la journée de travail. Il y a deux types d'ambulances, celles qui appartiennent à l'hôpital et celles qui appartiennent à la mairie de Bologne. Toutefois, l'hôpital préfère donner la priorité à sa propre flotte. Ce dernier service est géré par le département de transport de patients, la brancarderie, qui reçoit des demandes de transports de tous les pavillons et affecte les brancardiers aux véhicules disponibles. Une requête de transport est donc créée lorsqu'un patient doit se déplacer entre deux endroits sur le campus pour recevoir un service de soin. Chaque requête de transport reçue par le service

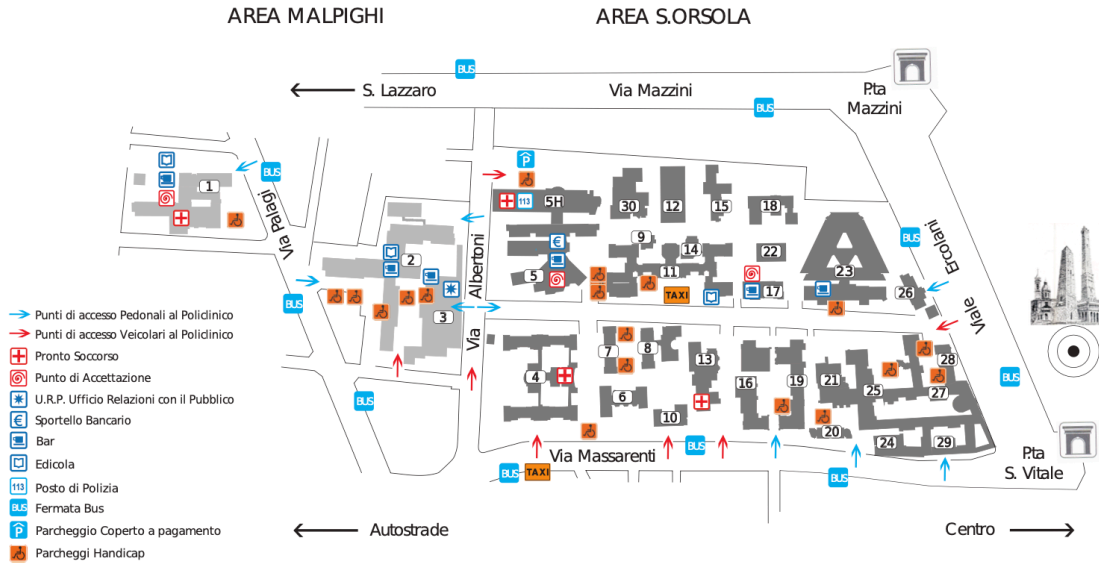


FIGURE 2.1 – Plan de l'hôpital universitaire Policlinico Sant'Orsola-Malpighi.

contient des informations sur le moment où le patient sera disponible, le temps du début de service médical pour ce client qui représente aussi le temps de livraison de patient au plus tôt, le temps de livraison de patient au plus tard, les pavillons et les étages où le patient doit être récupéré et ensuite livré, le nombre de places requises et le degré de priorité en fonction du degré d'urgence du cas à livrer. Si cette demande est connue depuis un certain temps, le service de transport peut être planifié à l'avance. Sinon, si la demande de transport arrive soudainement au cours de la journée, le service de transport doit être planifié en temps réel. La brancarderie reçoit des requêtes de transport 24h/24h. Cependant, le nombre de demandes au cours de la journée est plus important que pendant la nuit.

Même si notre travail adresse un cas réel bien spécifique, nous croyons que ce contexte représente généralement les pratiques et les situations typiques du monde hospitalier.

2.2 Hypothèses simplificatrices

Une étude approfondie de la base de données de l'hôpital a démontré que les demandes reçues au cours de la journée sont beaucoup plus élevées que les demandes reçues pendant la nuit. Alors, nous supposons qu'une journée de travail à l'hôpital est de 7h00 à 20h00. Aussi, puisque les demandes qui arrivent la nuit ne sont pas nombreuses, le service de la brancarderie peut les gérer facilement. Donc, les demandes reçues entre 20h00 et 7h00 sont éliminées.

En revanche, les demandes r qui ont une date de disponibilité t_r avant 7h00 et qui ont une fenêtre de temps qui commence après 7h00, c'est-à-dire $a_r \geq 7h00$, sont incluses dans notre planification et elles représentent les demandes planifiées à l'avance. Les autres demandes

qui arrivent au cours de la journée de travail ce sont des demandes imprévues et donnent un aspect dynamique à notre problème. En fait, environ 7.5% des demandes sont connues avant de commencer la journée de travail, tandis que la majorité des requêtes de transport (92.5%) sont imprévues et ont besoin d'être planifiées au moment de leurs arrivées en temps réel. Une autre chose qui peut augmenter la complexité de la planification, c'est le fait de recevoir des demandes pendant que les véhicules sont en train de faire des routes. Ceci rend la replanification des itinéraires obligatoire à chaque fois que de nouvelles demandes de transport arrivent.

Une autre caractéristique qui augmente la complexité du problème est liée aux deux types d'ambulances utilisés dans les hôpitaux. Ceux qui sont leur propriété et ceux qui sont propriété de la mairie de Bologne. Il n'y a aucune différence technique entre ces deux types (en termes de capacité ou d'équipements), mais l'hôpital préfère donner la priorité à ses véhicules pour des raisons de responsabilité. Dans notre problème, les ambulances seront considérées comme un seul type de véhicule.

2.3 Analyse de la situation actuelle

L'hôpital compte sur le service de brancarderie pour gérer les opérations de planification et de coordination de transport de patients. Ce service est constitué de deux personnes pour gérer et coordonner toutes les requêtes de transport reçues par le département, à noter, un opérateur et un assistant. Le processus de transport se déclenche lorsqu'un autre département, un service ou une unité de soins lance une requête de transport en l'introduisant dans un système d'informations les données du patients, l'origine et la destination, le nombre de places nécessaires, la fenêtre de temps et le niveau d'urgence. De son côté, le service de planification de transport reçoit les demandes sur un moniteur pendant que les brancardiers effectuent les itinéraires déjà conçus. Généralement, le planificateur assigne la nouvelle requête au prochain véhicule convenable et disponible. Il compte sur son expertise et son intuition et donc, il réalise les opérations d'affectation d'une façon totalement manuelle selon le principe du premier arrivé, premier servi, avec une attention particulière aux demandes urgentes.

Bien que la procédure de planification soit manuelle, elle est faite sur des ordinateurs, ce qui permet de garder une traçabilité de toutes les opérations de transport effectuées. En fait, l'hôpital possède une base de données de toutes les informations relatives aux systèmes de transport interne au fil des années. Dans le but de rendre la simulation du problème plus représentative, un extrait des informations de ces bases a été fait. Les données extraites contiennent environ 75 000 requêtes de transport qui couvrent toutes les opérations des brancardiers durant l'année 2016. Compte tenu du grand nombre d'informations reçu, une opération de nettoyage a été lancée. Toutes les informations inutiles au problème comme les données des patients et des brancardiers (nom et prénom) ont été supprimées. Le code et le nom de chaque

emplacement dans l'hôpital ont été modifiés.

Ces informations ont été utilisées pour créer une base de données qui représente les flux de transport interne de l'hôpital et ont été exportés dans des fichiers textes pour les utiliser dans la génération des instances.

Une analyse de ces données sera présentée dans la Section 2.4 suivante et les détails du générateur d'instance seront donnés dans le Chapitre 4.

2.4 Analyse des données

L'analyse des données fournies par l'hôpital est importante dans notre travail. En fait, une bonne compréhension de ces dernières mène à une meilleure maîtrise de ce qui se passe à l'intérieur de l'hôpital, ce qui nous aidera à bien modéliser et simuler ce cas. De plus, cette analyse nous aide à comprendre les flux de transport internes de l'hôpital. Ce qui permet de déterminer les origines et les destinations les plus fréquentées, et ainsi établir des stratégies de relocalisation. Aussi, l'étude statistique de ces données nous permet de faire des prévisions de la demande en déterminant sa distribution au cours d'une journée de travail. Cela nous permet de générer aléatoirement des scénarios qui peuvent être utilisés par l'approche stochastique.

Dans un contexte hospitalier, les demandes de transport des patients sont variables et difficiles à prédire. La Figure 2.2 présente une décomposition de la série chronologique des demandes durant l'année 2016. Dans le domaine de la statistique, une série chronologique est une série qui représente l'évolution d'un phénomène au cours du temps. Dans la suite de ce travail, nous considérons qu'une semaine de travail commence le dimanche et se termine le samedi. Une observation des requêtes reçues au cours de l'année 2016 montre une saisonnalité de la demande. Cette dernière fluctue entre un minimum de 73.7 demandes en moyenne durant les dimanches, et un maximum de 226.1 demandes en moyenne durant les lundis. Pour le reste de la semaine, la demande demeure proche de la borne supérieure avec une moyenne de 221.6, 211.3, 210.6 et 202.9 respectivement pour les mardis, mercredis, jeudis et vendredis. Tandis que pour les samedis, la demande moyenne baisse à 112.9 et s'approche de la borne inférieure du dimanche (voir la Figure 2.3). En règle générale, la demande est plus faible pendant les fins de semaine comparée aux autres jours.

Les courbes de l'erreur et de la tendance dans la Figure 2.2 montrent qu'il y a des jours particuliers durant l'année où nous observons une variation importante de nombre de demandes pour le même jour de la semaine. Par exemple, il existe seulement deux mardis de l'année 2016 où la demande est au-dessous de 160 requêtes (voir les Figures 2.4). Ces exceptions sont répétées pour d'autres jours de la semaine. Une investigation de ces cas révèle que ces variations sont liées à certains événements comme les fêtes nationales, les autres jours fériés de l'année et le manque de ressources lors des congés annuels des médecins. C'est aussi pour cette dernière raison que durant les fins de semaine il y a une baisse importante des demandes

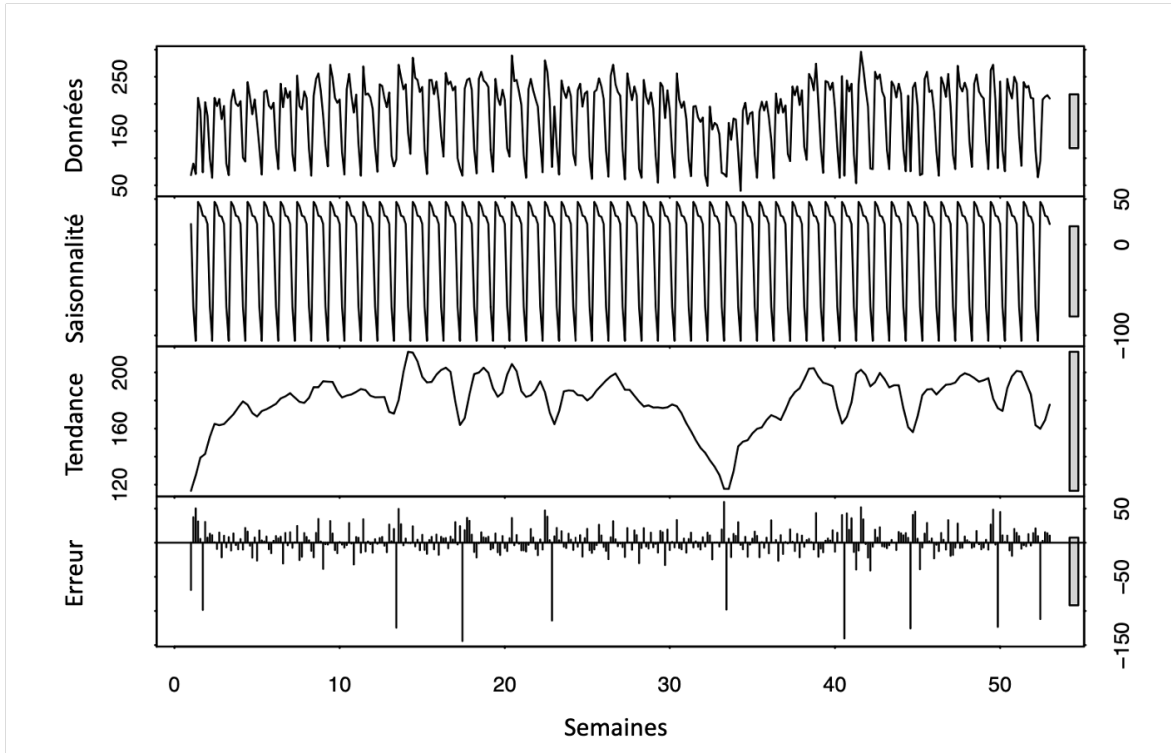


FIGURE 2.2 – Décomposition de la série chronologique des requêtes de transport reçues au cours de l'année 2016.

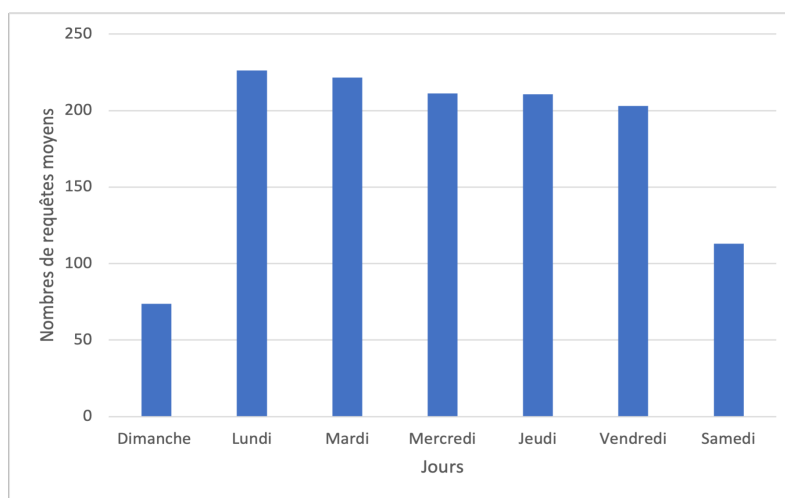


FIGURE 2.3 – Nombres de requêtes moyens par jour.

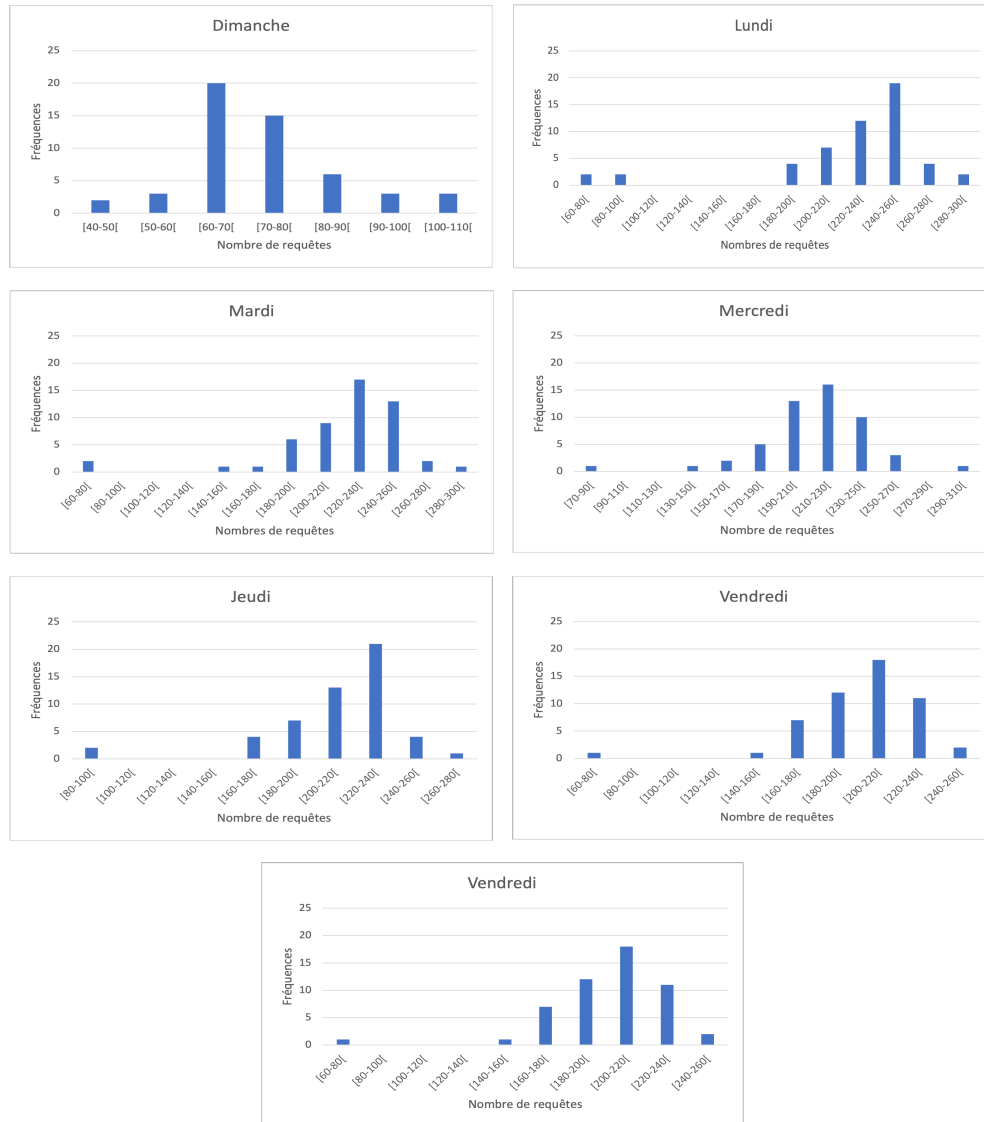


FIGURE 2.4 – Tendance annuelle des requêtes de transport pour chaque jour de la semaine.

de transport par rapport aux autres jours de la semaine. En fait, pour les jours fériés et les fins de semaine, tous les cas non urgents seront replanifiés pour un autre jour.

En termes de localisations, nous remarquons que plus de la moitié des demandes de transport émergent du pavillon d'urgence. Le premier étage de ce dernier pavillon, qui est représenté par le nœud "36" dans ce travail, représente 34.4% des origines des requêtes, à savoir, des lieux de cueillettes. Tandis que le deuxième étage de ce même pavillon, qui correspond au nœud "37", occupe 17.1% des lieux de cueillettes dans les demandes de transport. Ce pavillon d'urgence est aussi la destination la plus demandée dans les requêtes. Les nœud "37" et "36" occupent respectivement 17% et 10% des lieux de livraison demandés. La Figure 2.5 résume la fréquence des lieux de cueillettes et de livraisons pour l'année 2016.

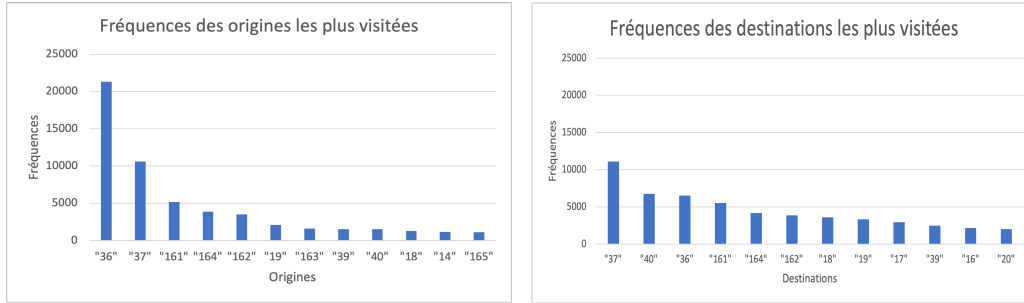


FIGURE 2.5 – Fréquence de visite des origines et des destinations.

Le Tableau 2.1 ci-dessous montre que pour chaque jour de la semaine, les requêtes de transport entre les deux premiers étages de l’urgence, nœuds "36" et "37", occupent la première place en termes des paires les plus visitées. Nous pouvons voir aussi que ce pourcentage augmente durant les fins de semaine vu que la majorité des médecins sont en repos. Le même phénomène peut être observé dans les jours fériés où le nombre de demandes de transport diminue et le grand pourcentage des demandes de transport sont effectué entre les étages de l’urgence. Généralement, les demandes reçues durant ces périodes sont des urgences et pas des consultations.

Tableau 2.1 – Les pairs de nœuds les plus visités pour chaque jour de la semaine

Dimanche		Lundi		Mardi		Mercredi		Jeudi		Vendredi		Samedi	
O-D	%	O-D	%	O-D	%	O-D	%	O-D	%	O-D	%	O-D	%
36-37	25.4	36-37	12.8	36-37	13.0	36-37	13.2	36-37	12.9	36-37	13.4	36-37	22.8
37-36	10.0	162-161	7.0	162-161	6.2	162-161	5.1	162-161	5.6	37-36	4.4	37-36	8.1
36-17	5.2	161-162	6.9	161-162	6.0	161-162	5.0	161-162	5.3	162-161	3.8	36-18	3.7
36-18	4.8	37-36	4.2	37-36	4.7	37-36	4.6	37-36	4.7	161-162	3.7	36-17	3.5
36-40	3.8	164-161	3.6	164-161	3.0	164-161	3.4	164-161	2.9	164-161	3.1	36-19	3.1
Somme	49.2		34.4		32.9		31.2		31.3		28.4		41.1

Durant une journée de travail, le nombre de requêtes varie d’une heure à une autre comme il est montré dans la Figure 2.6. Durant les premières heures de la journée, soit de 7h à 8h, la demande est à son minimum. Par la suite, elle commence à augmenter exponentiellement durant les prochaines heures jusqu’à atteindre son maximum entre 11h et midi. Ce phénomène est répété pour tous les jours de la semaine (du lundi à vendredi). En règle générale, 30% des demandes de la journée sont reçues entre 10h et 13h, l’intervalle qui représente les heures de pointe de la journée. Après 13h, le nombre de demandes diminue et la variation devient moins importante jusqu’à 20h. Durant les fins de semaine, le taux d’arrivée des requêtes change. Pour le dimanche, le pic des demandes est à 8h et la variation de la demande semble à être plus stable que le reste de la semaine. En fait, à partir de 8h, le nombre de requêtes varie entre un maximum de 7.4 à un minimum 4.6 demandes par heure. Pour le samedi, le nombre maximal des demandes est atteint à 10h. Les requêtes reçues entre cette heure est jusqu’à 13h représentent plus que 36% du nombre de requêtes total de la journée. Pour les

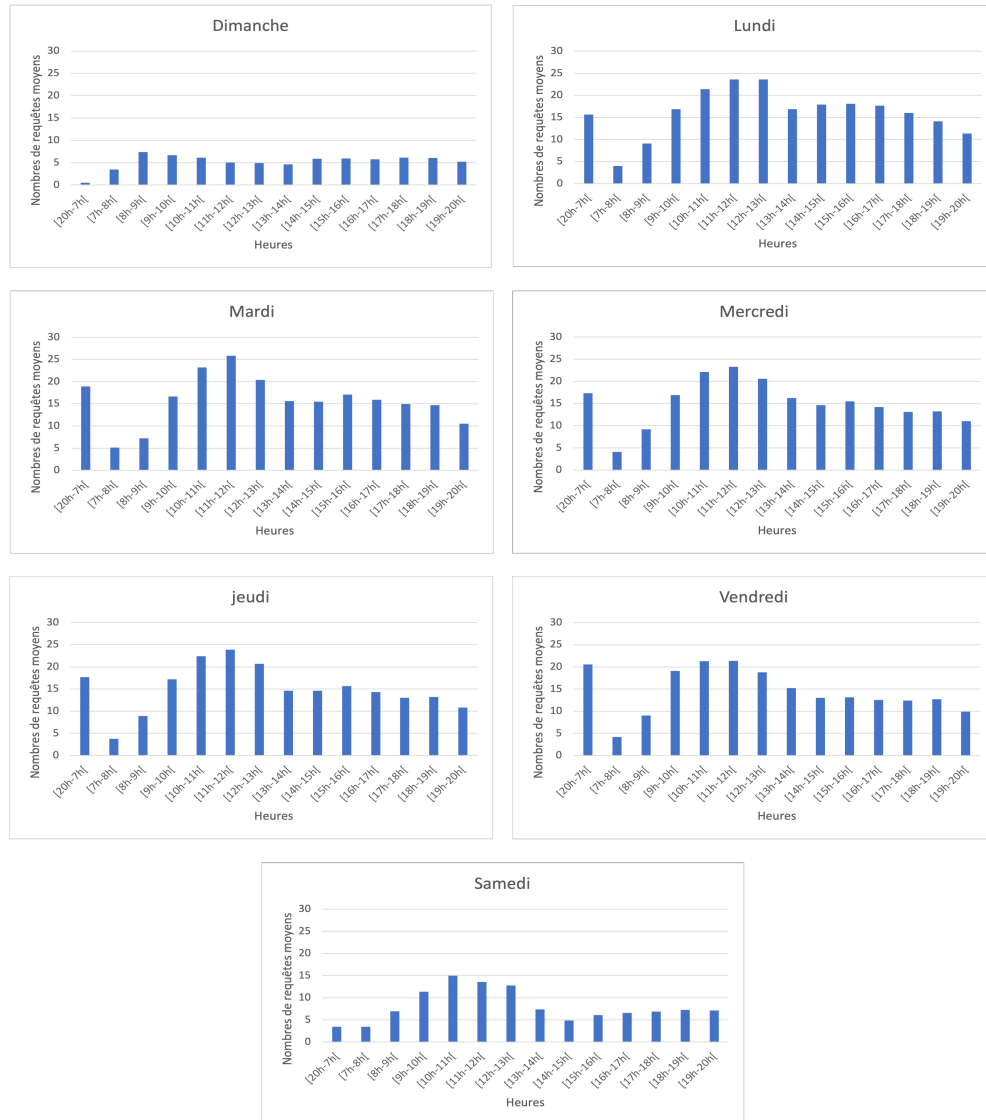


FIGURE 2.6 – Nombres de requêtes moyens par heures.

demandes de transport reçues entre 20h et 7h, elles sont généralement connues à l’avance pour les brancardiers. Le nombre de ces demandes est plus important du lundi au vendredi. En fait, il représente entre 7 à 10% du nombre total des demandes durant ces jours. Cependant, ce nombre diminue à 1% pour les dimanches et à 3% pour les samedis.

2.5 Modélisation

Le campus de l’hôpital universitaire de Bologne est composé de $P = \{1, 2, \dots, p\}$ pavillons dont chacun représente un service hospitalier distinct. En outre, chaque pavillon est composé de $F = \{1, 2, \dots, f\}$ étages. Toutes les combinaisons possibles de pavillon-étage sont représentées par N nœuds. La mission des brancardiers consiste à assurer le transport des patients entre les

pavillons et les étages. Pour ce faire, ils effectuent des itinéraires planifiés par la brancarderie en conduisant les véhicules entre deux points spécifiques $(i, j) \in A$ qui relient les nœuds les uns aux autres par des arêtes en créant un graphe complet $G = (N, A)$.

Soit $R = \{r_1, \dots, r_n\}$ l'ensemble de n requêtes de transport. Pour chaque demande $r \in R$ nous définissons une paire de nœuds (i, j) , où $i \in N$ et $j \in N$ sont respectivement le lieu de cueillette et le lieu de livraison du patient. Chaque nœud de cueillette ou de livraison est associé à une demande unique, ce qui permet d'avoir plusieurs nœuds pour une seule localisation. En outre, chaque requête $r \in R$ est reçue à une date t_r appelée aussi la date de disponibilité, une fenêtre de temps souple $[a_r, b_r]$ où a_r représente le temps du début de service au plus tôt qui est aussi la date de disponibilité du patient pour la cueillette et b_r représente le temps du début de service au plus tard qui indique le dernier moment où la livraison peut être effectuée sans aucun retard. Aussi, pour chaque demande, le niveau d'urgence est exprimé par un poids w_r , et l'espace nécessaire pour transporter le patient dans le véhicule est indiqué par un poids p_r . Au début d'une journée de travail, la plupart des demandes de transports sont inconnues. Le voile sur les demandes se lève au fur et à mesure que le temps passe, en d'autres termes, elles sont dévoilées dynamiquement au cours de la journée.

En ce qui concerne les véhicules, l'hôpital a mis en service une flotte hétérogène $K = \{1, \dots, k\}$ pour assurer le transport des patients. Cette flotte est composée de deux types de véhicules, dont deux ambulances et trois véhicules électriques. En fonction du type q , chaque véhicule a une capacité c_{qj} avec $j = \{1, \dots, k\}$, et $q = \{1, 2\}$. La capacité de transport des ambulances est limitée à un seul patient. Du coup, ce type de voiture est limité à des trajets entre un seul lieu de cueillette et un seul lieu de livraison. Toutefois, les véhicules électriques n'ont pas ces restrictions ce qui leurs permettent de transporter jusqu'à quatre patients à la fois. Ce qui donne plus de flexibilité en les permettant d'effectuer des trajets multiples entre plusieurs lieux de cueillette et plusieurs lieux de livraison. La flotte est disponible pendant toute la journée de travail. Les changements de personnel sont pris en compte par la brancarderie. Du coup, ils ne provoquent aucun retard pendant le quart de travail. Chaque véhicule commence et termine ses itinéraires à partir d'un dépôt central $0 \in N$ situé au premier étage du pavillon d'urgence de l'hôpital. Les véhicules ont la possibilité de se relocaliser et d'attendre ailleurs pendant le quart du travail.

L'ensemble des arêtes A est défini comme suivant $A = \{(i, j) : i, j \in N, i \neq j\}$. Chaque arête $(i, j) \in A$ nécessite un temps de trajet $t_{i,j}^q$, où $q = \{1, 2\}$, qui dépend du type de véhicule. Les ambulances font des trajets externes qui relient les pavillons. Tandis que les véhicules électriques sont utilisés principalement dans des tunnels souterrains, mais peuvent aussi être utilisés sur les routes externes comme les ambulances.

L'objectif de ce problème est de concevoir un ensemble d'itinéraires dans le but de satisfaire les demandes de transport des patients. Les fenêtres de temps sont définies comme souples

afin de tolérer les retards. Ainsi, le retard l_r de la requêtes r est défini comme la différence entre τ_r le temps de fin de livraison pour la requête r et b_r le dernier moment où on peut livrer le patient sans retard. Évidemment, ce retard n'est considéré que lorsque sa valeur est positive :

$$l_r = \max\{0, \tau_r - b_r\}.$$

En plus d'engendrer une insatisfaction des patients, chaque retard a un impact direct sur le planning du personnel de l'hôpital et surtout sur les médecins qui sont en contact direct avec les patients. L'objectif est d'optimiser le transport des patients en minimisant la distance totale parcourue par les brancardiers et la somme pondérée des retards. La pondération du retard d'une requête dépend de son niveau d'urgence w_r et d'une constante β utilisée pour donner au retard une plus grande importance que la distance. L'objectif peut s'écrire ainsi :

$$\sum_{p \in L} \sum_{(i,j) \in p} d_{ij} + \sum_{r \in R} \beta w_r l_r$$

où L est l'ensemble de routes et (i, j) sont les arcs construisant la route p de l'ensemble L .

Pour qu'une solution soit réalisable, elle doit satisfaire l'ensemble de contraintes suivant : chaque demande de transport doit être servie (1), chaque itinéraire commence et se termine au dépôt (2), la capacité de véhicules doit être respectée en tout temps (3), les opérations de cueillette et de livraison d'une demande doivent être effectuées par le même véhicule (4), chaque cueillette doit être faite avant de faire sa propre livraison (5), les véhicules ne peuvent pas sortir pour faire une cueillette d'un patient relatif à une demande r avant sa date de disponibilité t_r (6), pour une requête r , le patient ne peut pas être cueilli avant a_r , mais si le véhicule arrive avant ce temps, il peut attendre (7).

Il est important de mentionner que ce problème de cueillettes et de livraisons dynamique étudié dans ce mémoire ne contient pas la contrainte de la durée maximale du trajet. De ce fait, notre problème ne peut pas être classé comme un vrai problème de transport à la demande (DARP). L'absence de cette contrainte est justifiée par les fenêtres de temps qui sont très courtes et la capacité réduite des véhicules, et en conséquence, les patients restent dans les véhicules une durée relativement courte.

Chapitre 3

Algorithmes et approches de résolution

Introduction

Dans ce chapitre, nous présenterons les différentes approches de résolutions développées. La première partie présentera notre méthodologie de résolution. Ensuite, la deuxième partie décrira l'approche statique en mettant l'accent sur l'heuristique de la recherche adaptative à grand voisinage. La troisième partie illustrera l'approche dynamique en définissant la procédure de résolution, les différents événements, les stratégies d'attente et de relocalisation et l'heuristique de réoptimisation utilisée. Enfin, la dernière partie présentera notre approche stochastique en se focalisant sur l'approche de planification basée sur des scénarios échantillonnés et la fonction de consensus.

3.1 Méthodologie de résolution

Dans ce mémoire, nous avons développé trois approches de résolutions différentes. Chacune de ces approches est constituée d'un ensemble d'algorithmes et d'heures permettant d'aborder le problème d'un point de vue différent.

La première approche est un algorithme qui permet de résoudre le problème dans sa version statique. Cette approche représente l'état utopique du problème où toutes les informations sont connues.

La deuxième approche aborde le problème dans sa version dynamique. Pour ce faire, un algorithme dynamique qui utilise une heuristique de réoptimisation est développé. Ce dernier ne prend pas en compte les connaissances futures sur les demandes ou les prévisions des événements à venir. L'objectif de cette approche est de fournir au département de transport interne de l'hôpital un outil d'aide à la décision en temps réel. Ce qui permet également de résoudre

le problème dans les pires conditions. Cet algorithme permet de construire les itinéraires pour les brancardiers en temps réel sans prendre en considération aucune information à l'avance. C'est-à-dire, au cours de la journée, et au moment qu'une demande est révélée, l'algorithme est capable d'insérer cette nouvelle demande et de mettre à jour les itinéraires déjà établis. Pour améliorer la performance de cette approche, une stratégie d'attente et deux stratégies de relocalisation ont été développées. Ces trois stratégies ont été utilisées pour créer trois versions différentes de cet algorithme. Cela nous donne l'opportunité de tester et de comparer les résultats afin de trouver la meilleure stratégie de relocalisation.

La troisième approche considère également le problème dans sa version dynamique, mais en tenant compte des prévisions et des connaissances futures sur les événements à venir. L'objectif de cette approche est de traiter le problème dans des conditions réelles où le planificateur construit les itinéraires en temps réel sans aucune connaissance sur les événements à venir. Il se base sur son expérience avec les demandes passées afin d'optimiser les flux de transport. Cette approche de planification est donc basée sur des scénarios échantillonnés générés suivant des lois de probabilités. De plus, nous avons développé une nouvelle fonction de consensus dans le but de sélectionner une relocalisation pour chaque véhicule en utilisant des données stochastiques. Cette dernière est ainsi considérée comme une troisième stratégie de relocalisation. Comme dans l'approche dynamique, nous avons intégré la stratégie d'attente et les deux stratégies de relocalisation dans cette approche. Donc, nous avons obtenu trois versions de l'algorithme.

Dans ce mémoire, tous les algorithmes utilisent l'ALNS pour résoudre les problèmes de cueillettes et de livraisons.

Notre procédure de résolution compte principalement sur une étape finale de simulation. Pour chaque approche, nous avons développé un simulateur qui utilise les instances comme une entrée. Au début, une fonction charge toutes les informations relatives à la journée de travail sélectionnée. Ensuite, elle crée les nœuds de cueillettes et de livraisons sous format de paires. Chacun de ces nœuds créés contient toutes les informations relatives à une demande comme la date de disponibilité, la fenêtre de temps, le pavillon, l'étage, le type de ce nœud (nœud de cueillette ou de livraison), le nœud de cueillette ou de livraison associé à ce nœud, un identifiant unique et un chiffre qui représente la combinaison du pavillon et de l'étage de la demande. Cette fonction charge également les matrices de distances et de temps entre les différents endroits de l'hôpital. Ensuite, une autre fonction crée les chauffeurs selon le nombre d'ambulances et de véhicules électriques disponibles et génère les demandes de transport. La génération des demandes dans les trois approches se fait de manière différente. Finalement, une simulation de la journée de travail est lancée. La simulation appelle les différents algorithmes et heuristiques pour gérer ces demandes et construire les itinéraires pour les brancardiers. La simulation nous permet également d'avoir un modèle de l'hôpital, de son environnement et de la situation actuelle de ses flux et de ses opérations de transport. Ce qui nous donne

l'opportunité de tester et d'évaluer les différentes approches avec ce modèle.

Les sections suivantes seront donc consacrées à explorer les détails de chaque approche telles que la procédure de résolution et les pseudo-codes des algorithmes et des heuristiques.

3.2 Approche statique

Nous commençons par étudier le problème dans sa version statique. Ce cas suppose que nous connaissons d'avance toutes les requêtes qui auront lieu. Cette version du problème représente le cas où nous avons les meilleures conditions possibles. Les résultats obtenus par cette approche sont les meilleurs résultats possibles et ils représentent une borne inférieure sur les retards et les distances parcourues. Toutefois, il faut dire que cette approche ne représente pas le cas réel.

3.2.1 Procédure de résolution

La procédure commence par l'appel de la fonction qui charge l'instance du jour sélectionné et crée les différents nœuds. Ensuite, une simulation de ce jour est lancée. L'algorithme de simulation appelle la fonction qui génère toutes les requêtes de transport de ce jour. Chaque requête de transport est associée à deux nœuds uniques de cueillette et de livraison. Puis, l'heuristique d'ALNS est exécutée pour assurer l'optimisation.

3.2.2 Recherche Adaptative à Grand Voisinage

L'heuristique d'ALNS fut introduite par Ropke et Pisinger (2006) est une approche de résolution utilisée dans la littérature pour résoudre les problèmes de cueillettes et de livraisons avec des fenêtres de temps. Ce qui distingue cette heuristique par rapport aux autres approches classiques, c'est sa capacité de modifier de 30 à 40% de la solution à chaque itération (Ropke et Pisinger, 2006). Contrairement à cette dernière, les approches classiques ne sont pas capables de faire autant de modifications sur une solution. Généralement, elles font que de petites modifications de sorte que la recherche n'explore qu'une petite partie de l'espace des solutions. En revanche, l'idée de base de l'ALNS est de corriger ce point de faiblesse avec de grands mouvements de façon que la recherche locale peut aller d'une zone de l'espace de solution à une autre. Et donc, elle peut explorer des zones loin de la solution actuelle.

L'algorithme de l'ALNS est basé sur le principe de destruction et de reconstruction. Les opérations de destruction et de reconstruction sont réalisées avec des opérateurs de retrait et d'insertion définis par les auteurs. Ces opérateurs ne sont que des heuristiques de retrait et d'insertion.

Un aperçu général de l'heuristique est donné dans l'Algorithme 1 ci-dessous.

Avant de lancer l'ALNS, nous commençons par créer une solution initiale s dont la qualité n'est pas jugée importante à ce stade, étant donné qu'une grande partie de celle-là sera détruite dès les premières itérations de cet algorithme. Nous avons utilisé une heuristique d'insertion séquentielle afin de créer la solution initiale. Ensuite, cette dernière est copiée dans la solution optimale s^* . Par la suite, l'ALNS est lancée pour un nombre d'itérations η à l'aide de plusieurs opérateurs de suppression et d'insertion. Ceci représente l'une des caractéristiques majeures de l'ALNS. La recherche lance une séquence de suppression puis de réinsertion des demandes de transports dans la solution. Pour ce faire, un nombre entier de requêtes q à retirer puis à réinsérer est sélectionné aléatoirement à chaque itération. Par la suite, un opérateur de retrait et un opérateur d'insertion sont sélectionnés à chaque itération. Une copie s' de la solution s est créée et l'opérateur de retrait choisi retire les q demandes sélectionnées de la solution s' . Ces requêtes retirées sont enregistrées dans une liste qui est utilisée directement après les opérations de suppression par l'opérateur d'insertion sélectionné afin de réinsérer les q requêtes et reconstruire la solution s' . Par la suite, une étape de vérification est nécessaire. En fait, si la solution s' satisfait le critère d'acceptation déjà défini, elle sera acceptée. Sinon, elle sera refusée. Si s' est acceptée, elle devient alors la nouvelle solution courante s . En plus, si cette dernière solution a un retard total pondéré plus petit que la meilleure solution s^* , alors elle est copiée dans s^* . Après η itérations, la meilleure solution visitée est retournée.

Algorithme 1 : L'heuristique de l'ALNS

```

1: Construire une solution initiale  $s$ 
2:  $s^* = s$  et initialisation des poids
3: pour  $i \leftarrow 1$  jusqu'à  $\eta$  faire
4:   Générer un nombre aléatoire  $q$  de demandes à retirer
5:   Sélectionner aléatoirement un opérateur de suppression et un opérateur d'insertion
6:    $s' = s$ 
7:   Retirer  $q$  demandes de  $s'$  avec un opérateur de suppression sélectionné
8:   Insérer les demandes non planifiées dans  $s'$  avec un opérateur d'insertion sélectionné
9:   si  $s'$  satisfait un critère d'acceptation alors
10:      $s = s'$ 
11:     si  $s'$  est meilleur que  $s^*$  alors
12:        $s^* = s$ 
13:     fin si
14:   fin si
15:   Ajuster les poids
16: fin pour
17: retourner  $s^*$ 

```

Les opérateurs de retrait sont des heuristiques qui détruisent des éléments de la solution actuelle en retirant un nombre de demandes q déjà choisi. La sélection de ces requêtes se fait selon un critère donné. Dans notre approche, nous avons utilisé l'heuristique de retrait aléatoire et l'heuristique de retrait corrélé présenté par Ropke et Pisinger (2006) comme des

opérateurs de retrait.

Comme son nom l'indique, l'opérateur de retrait aléatoire retire les q demandes de transport d'une façon aléatoire. Toutes les requêtes qui se trouvent dans la solution ont la même probabilité d'être retirées. Donc, à chaque itération, quand le nombre aléatoire généré correspond à l'indice de l'un des nœuds de cueillette ou de livraison d'une demande de transport, cette dernière est retirée. Tous les nombres aléatoires générés sont des entiers qui se trouvent dans l'intervalle $[1, n]$.

L'opérateur de retrait corrélé est connu sous le nom anglais "Related Removal" proposé la première fois par Shaw (1997). L'idée de ce dernier consiste à trouver une relation de corrélation entre les différents éléments d'une solution. Ensuite, l'opérateur retire les éléments similaires de la solution dans le but de les réinsérer, de manière plus optimisée, avec l'opérateur d'insertion. Dans notre approche, nous avons considéré la distance entre les nœuds comme une métrique de corrélation. Alors, la suppression des requêtes se fait avec un opérateur de retrait basé sur la distance. Le pseudo-code de cet opérateur est donné dans l'Algorithme 2.

Algorithme 2 : L'heuristique de retrait corrélé

```

1: fonction : RetraitCorrélé( $s \in \{Solutions\}$ ,  $q \in \{\mathbb{N}\}$ ,  $p \in \{\mathbb{R}_+\}$ )
2: Sélectionner une requête aléatoire  $r$  de  $s$ 
3: Insérer  $r$  dans l'ensemble de demande  $D$  :  $D \leftarrow \{r\}$ 
4: Retirer la demande  $r$  de  $s$  :  $s = s \setminus \{p(r), d(r)\}$ 
5: tant que  $|D| < q$  faire
6:   Sélectionner une requête aléatoire  $r$  à partir de l'ensemble  $D$ 
7:   Créer un tableau vide  $L$ 
8:   pour toutes requêtes  $i \in s$  faire
9:      $b_i \leftarrow c_{p(r)p(i)} + c_{d(r)d(i)}$ 
10:     $L \leftarrow L \cup \{i\}$ 
11:   fin pour
12:   Trier  $L$  selon les valeurs  $b_i$ 
13:   Sélectionner un nombre aléatoire  $y$  dans l'intervalle  $[0,1]$ 
14:    $pos \leftarrow |L| \cdot y^p$ 
15:   Sélectionner la requête  $i$  à la position  $pos$  dans  $L$ 
16:    $D \leftarrow D \cup \{i\}$ 
17:    $s \leftarrow s \setminus \{p(i), d(i)\}$ 
18: fin tant que
19:   Retourner  $D$ 

```

L'algorithme commence par sélectionner aléatoirement une requête r . Cette requête est insérée dans une liste D puis elle est retirée de la solution s . L'étape suivante consiste à lancer une boucle pour répéter les étapes 6 à 17 jusqu'à ce que les q demandes sont retirées de la solution s . Dans cette boucle, la procédure débute par choisir aléatoirement une requête r de la liste D . Toutes les requêtes i de la solution s sont triées dans un tableau L . Le tri se fait selon les coûts b_j . Pour deux requêtes r et i , ce coût n'est que la somme de la distance entre leurs

nœuds de cueillettes respectivement $p_{(r)}$ et $p_{(i)}$ et la distance entre leurs nœuds de livraison respectivement $d_{(r)}$ et $d_{(i)}$ (voir l'étape 9). Ensuite, une requête est choisie du tableau L trié selon l'ordre de corrélation de la demande r avec toutes les autres demandes i de la solution s en considérant la distance comme métrique. Ce choix de requête se fait à travers une méthode qui inclut l'aspect aléatoire, mais en même temps elle biaise vers les requêtes qui ont les plus petits b_i . En fait, le paramètre p est utilisé pour contrôler la diversification. Avec une valeur proche de 1, la corrélation est un peu ignorée et le critère de choix devient plutôt aléatoire, ce qui est un peu similaire à l'opérateur de retrait aléatoire. En revanche, le plus le paramètre p est grand, le plus la valeur de y^p tend vers zéro, ce qui mène à sélectionner des demandes proches de r . À la fin, une liste avec toute les q requêtes à retirer est retournée.

Les opérateurs d'insertion sont des heuristiques qui réinsèrent les requêtes retirées par les opérateurs de retrait de manière à reconstruire la solution. Généralement, ces heuristiques d'insertion sont exécutées sur deux étapes. La première étape correspond à la sélection du nœud à insérer, tandis que la deuxième étape consiste à exécuter la méthode d'insertion. Dans notre approche, nous avons utilisé l'opérateur d'insertion séquentielle et l'opérateur d'insertion avec regret.

L'heuristique d'insertion séquentielle utilisée insère les requêtes d'une manière qui minimise le détour et ainsi la fonction objectif. Cette dernière est une fonction de coût qui dépend de la distance parcourue et du retard total pondéré. À titre d'exemple, si une cueillette (i) est insérée entre le nœud (j) et son successeur $succ(j)$ et que le nœud de livraison $l(i)$ associé au nœud (i) est inséré entre le nœud (k) et son successeur $succ(k)$ alors notre détour δ_{cout} est :

$$\delta_{cout} = c_{j,i} + c_{i,succ(j)} - c_{j,succ(j)} + c_{k,l(i)} + c_{l(i),succ(k)} - c_{k,succ(k)}.$$

Les requêtes sont traitées une à la fois. Chaque insertion faite doit respecter les contraintes de notre problème comme les contraintes de capacité et les contraintes de précédence qui stipulent qu'un nœud de cueillette doit être visité avant son nœud de livraison pour la même demande. Malgré l'efficacité de cette méthode, elle ne permet pas de connaître si une insertion aura un impact sur les prochaines insertions ou pas.

En revanche, l'heuristique d'insertion avec regret, présentée par Potvin et Rousseau (1995), peut faire face à ce problème. L'heuristique définit une valeur appelée la valeur de regret. Cette dernière n'est que la différence entre le coût d'insertion d'une requête dans son meilleur itinéraire et son coût d'insertion dans son deuxième meilleur itinéraire. À chaque itération, la demande avec la plus grande valeur de regret est sélectionnée pour être insérée dans sa meilleure position et donc, dans la position qui minimise la fonction objectif. Ropke et Pisinger (2006) suggèrent que nous devons choisir l'insertion qui nous mène à regretter le plus si elle n'est pas faite tout de suite. Encore, une autre extension de cette heuristique est utilisée dans notre approche. C'est l'heuristique avec k -regret utilisé dans le travail de Ropke et Pisinger

(2006). En utilisant cette dernière, nous avons plus de possibilités à insérer les requêtes dans les itinéraires. Plus précisément, une heuristique avec k -regret calcule le coût d'insertion d'une requête dans les k meilleurs itinéraires. Et elle choisit l'itinéraire avec la valeur de regret la plus grande entre le meilleur itinéraire et les $k-1$ meilleurs itinéraires.

Après avoir présenté les opérateurs, maintenant, il faut expliquer comment ces opérateurs sont sélectionnés. L'étape de sélection des opérateurs est très importante vu qu'elle a un impact sur la façon de rechercher et sur la façon d'explorer l'espace des solutions. La performance des opérateurs est liée aux types de problèmes à résoudre. De ce fait, l'ALNS utilise un mécanisme intelligent pour choisir les opérateurs. À chaque itération, les opérateurs de retrait et d'insertion sont sélectionnés automatiquement en se basant sur des statistiques des itérations précédentes. C'est grâce à cette caractéristique que la recherche dans cet algorithme est adaptative. Le mécanisme de sélection automatique utilise des poids qui sont associés à chaque opérateur. Au début, tous les poids sont initialisés à 1. À chaque itération, une évaluation est faite. Si un opérateur est capable d'améliorer les solutions, son score est mis à jour. À chaque itération, l'heuristique sélectionnera les opérateurs ayant le score le plus élevé et après un nombre d'itérations, réinitialisera les poids une autre fois à 1 afin d'éviter le cas où des poids deviendront très grands et seront toujours choisis.

L'algorithme de l'ALNS trouve une nouvelle solution à chaque itération. Cependant, cette solution ne sera pas forcément acceptée. La solution trouvée doit être évaluée avec un algorithme d'évaluation et d'acceptation qui considère des critères d'acceptation similaires à ceux utilisés dans le travail de Ropke et Pisinger (2006). Le critère d'acceptation peut être simple, comme par exemple, une solution n'est acceptée que lorsqu'elle est meilleure que la solution actuelle. Selon les auteurs, un simple critère comme celui-là peut nous piéger dans un minimum local. Pour affronter ce problème, ils proposent d'accepter occasionnellement des solutions moins bonnes que la solution actuelle. Dans notre travail, l'algorithme d'acceptation utilise une technique similaire à celle du recuit simulé. En fait, il existe deux cas de figure. Le premier c'est le cas où la nouvelle solution trouvée s' a une valeur $f(s')$ meilleur que $f(s)$ la valeur de la solution actuelle s . Dans ce cas, la solution s' est acceptée. Le deuxième cas c'est lorsque la nouvelle solution trouvée s' est pire que la solution actuelle s . Alors, cette solution peut être acceptée avec une probabilité $e^{-\frac{(f(s')-f(s))}{T}}$ avec $T > 0$, où T est un paramètre appelé la température. La température a une valeur initiale T_{init} , ensuite, à chaque itération T est calculé en utilisant la formule $T = T \cdot c$ où c est $0 < c < 1$ représente le taux de refroidissement. Puisque la valeur de c est entre 0 et 1, alors la valeur de la température diminue avec chaque itération. Ce qui réduit également la probabilité d'accepter une solution de qualité inférieure. La température initiale est calculée en utilisant la solution initiale s et un paramètre $z' : f(s)(1 + z)$.

Il faut dire que notre seule modification dans l'algorithme de l'ALNS se résume dans le nombre d'itérations η et les opérateurs utilisés (nous n'utilisons pas tous les opérateurs). De

plus, nous pouvons ajouter que dans notre problème nous avons des fenêtres de temps souples, et donc, nous pouvons toujours trouver une solution réalisable. Donc, à la fin de chaque étape d'optimisation dans l'algorithme, toutes les requêtes de transport sont affectées à un véhicule.

3.3 Approche dynamique

Dans la littérature, un problème est dit dynamique s'il évolue et change au cours du temps. Dans notre cas, ceci signifie qu'au début de la journée, notre problème ne contient que les demandes qui ont une date de disponibilité avant 7h00. D'ailleurs, tout au long d'une journée de travail, aucune information sur les requêtes de transport n'est donnée. Puis, au cours de la journée, à chaque fois qu'un service de soins au sein de l'hôpital a besoin des brancardiers pour effectuer un service de transport, une demande de transport doit être soumise. Par conséquent, le problème évolue et change en recevant des requêtes de transport au fur et à mesure que le temps passe. Contrairement à l'approche statique où toutes les informations sur les demandes de transport sont connues avant de lancer la simulation, la nouvelle approche développée, qui est une approche dynamique, permet de faire une simulation représentative de la situation réelle des flux de transport dans l'hôpital. Dans cette approche dynamique, aucune information sur les demandes de transport n'est prise en compte dans la simulation, à l'exception des demandes planifiées et qui représentent environ 7.5% de la demande totale. C'est pourquoi toutes les informations deviennent totalement connues seulement lorsque la simulation atteint le dernier événement de la journée.

3.3.1 Procédure de résolution

La procédure commence par le chargement de l'instance du jour et la création des nœuds, tous comme l'approche précédente. Ensuite, la simulation est exécutée. Mais cette fois, l'algorithme ne génère pas toutes les demandes de transport de la journée. Il génère que les demandes qui ont une date de disponibilité inférieure ou égale à l'instant t actuel de l'algorithme. Autrement dit, il génère toute la partie de l'instance connue jusqu'à l'instant courant de l'algorithme. Cette partie de l'instance représente "le scénario réel". Dans cette approche, nous utilisons également une heuristique de réoptimisation. Donc, au début de la simulation, cette heuristique construit les itinéraires en utilisant les demandes disponibles jusqu'à l'instant t uniquement. Les cueillettes et les livraisons qui ont été exécutées sont verrouillées dans la solution et ne peuvent plus être modifiées. Puis, la simulation met à jour le temps actuel t de l'algorithme pour qu'il puisse générer encore toutes les demandes disponibles avant t . Ces nouvelles demandes générées seront ensuite insérées dans les itinéraires. Cette procédure est répétée à chaque itération jusqu'à la dernière requête de transport.

3.3.2 La gestion des événements

Au cours d'une journée de travail, au fil du temps, nous avons de nouvelles requêtes de transport qui sont révélées et des opérations de cueillettes et de livraisons qui ont été exécutées ou qui sont en cours. Ces événements augmentent le caractère dynamique de notre problème et rendent la prise de décision plus complexe.

Contrairement au cas statique où nous avons la possibilité de modifier les décisions passées, présentes et futures. Dans l'approche dynamique, nous n'avons plus cette option puisqu'il s'agit d'une simulation de la situation réelle des flux de transport. Dans ce cas, chaque itinéraire doit être conçu avec le temps qui s'écoule. Par conséquent, à chaque événement qui se produit, une décision doit être prise en temps réel. Dans notre approche dynamique, nous autorisons les deux types de décisions suivant :

- Les décisions d'affectation : ce sont les décisions classiques qui consistent à affecter les demandes aux chauffeurs (les brancardiers) afin de construire les itinéraires. Ces décisions doivent contenir toutes les informations de cette affectation telle que l'identifiant du brancardier, les nœuds de cueillette et les nœuds de livraison ainsi que les informations relatives aux patients (urgence, nombre de places requises).
- Les décisions stratégiques : ce sont les décisions qui sont effectuées dans le but d'optimiser la fonction objectif et n'impliquent aucune demande de transport. Ces décisions concernent des stratégies d'attente et de relocalisation. Elles dictent si un véhicule doit attendre dans son emplacement actuel ou attendre dans un autre emplacement jusqu'à la réception d'une nouvelle affectation.

Chaque décision prise doit être enregistrée afin d'assurer la traçabilité des déplacements et des trajets effectués par les véhicules jusqu'à l'instant actuel de l'algorithme ou afin de concevoir les déplacements et les trajets à effectuer dans le moment présent. En fait, dans notre approche dynamique, à chaque fois qu'un nouvel événement arrive, l'algorithme collecte toutes les informations connues jusqu'à cet instant, et il réexécute une étape de réoptimisation. Dans ce mémoire, nous considérons les trois événements suivants :

- La révélation d'une nouvelle requête de transport et au moins un véhicule est disponible.
- La fin de l'opération de livraison pour un véhicule.
- L'arrivée d'un véhicule qui a été relocalisé au nouvel emplacement.

Tout au long de la simulation, et à chaque fois qu'un événement se produit, la solution est mise à jour en se basant sur les décisions qui ont été prises et sur les nouvelles demandes qui sont arrivées. Toutes les décisions qui ont été prises avant l'occurrence de l'événement sont enregistrées et représentent les parties exécutées de la solution. Ensuite, à chaque étape d'optimisation, ces décisions sont utilisées pour reconstruire les parties de la solution exécutée

avant l'occurrence de l'événement et sont fixées de manière permanente dans la solution. Par conséquent, toutes les cueillettes et les livraisons et toutes les opérations d'attente ou de relocalisation qui ont été effectuées ou qui sont en cours d'exécution sont définitives et ne peuvent plus être modifiées. Dans le but de prendre de nouvelles décisions relatives aux nouvelles informations révélées, une opération de réoptimisation est lancée.

3.3.3 Les stratégies d'attente et de relocalisation

À chaque événement qui se produit, l'algorithme génère toutes les demandes connues jusqu'à l'instant présent t . Ensuite, en utilisant les décisions enregistrées, l'algorithme reconstruit les déplacements précédents et il exécute une étape de vérification de l'état de chaque chauffeur. En fonction de l'état du chauffeur à l'instant t , l'algorithme prend une décision. Pour tous les brancardiers qui ont terminé une livraison et qui n'ont pas d'autres affectations à faire, ils seront relocalisés.

L'idée des stratégies de relocalisation consiste à relocaliser un brancardier, qui a terminé sa dernière livraison, à un endroit différent. L'objectif est d'anticiper les déplacements potentiels dans le futur afin d'optimiser la fonction objectif. En fait, si un chauffeur n'est affecté à aucune demande, ça sera plus avantageux de le rapprocher de la source potentielle de sa prochaine demande. Ceci diminuera le temps de trajet ainsi que la distance parcourue par le brancardier pour arriver à sa prochaine cueillette lorsqu'il reçoit une nouvelle affectation. Également, ceci implique de meilleurs résultats puisque la fonction de coût de l'algorithme se base toujours sur la dernière position du brancardier avant de décider sa prochaine destination.

En se basant sur l'analyse des données présentées dans le chapitre 2, nous avons développé la stratégie d'attente et les deux stratégies de relocalisation suivantes :

- Stratégie 1 : c'est une stratégie d'attente où chaque véhicule qui n'est pas affecté à une requête reste où il est jusqu'à ce qu'il soit affecté de nouveau à une nouvelle requête. Si un véhicule est affecté à une nouvelle requête, il doit se déplacer pour l'effectuer.
- Stratégie 2 : celle-ci vise à relocaliser chaque véhicule qui n'est pas affecté à une demande doit se déplacer vers l'urgence et attendre d'être affecté à une nouvelle requête.
- Stratégie 3 : cette stratégie de relocalisation qui est similaire à la stratégie 2, mais après la dernière livraison, au lieu de se déplacer vers l'urgence, chaque véhicule doit se déplacer au rez-de-chaussée du dernier pavillon de livraison.

En termes d'effort de calcul, la première stratégie représente le meilleur choix. En effet, il n'y aura aucune décision additionnelle à inclure dans la solution et l'ALNS basera ses calculs sur la dernière position de livraison. Cependant, cela pourrait augmenter le temps de déplacement du brancardier vers son prochain lieu de cueillette, ce qui peut être expliqué par le temps supplémentaire nécessaire pour qu'un véhicule puisse quitter l'étage ou le pavillon de son lieu

d'attente actuel vers sa prochaine destination qui ne sera pas probablement dans le même étage ou dans le même pavillon.

La deuxième stratégie représente la stratégie actuellement utilisée par l'hôpital. Cette stratégie est adoptée parce que la plupart des requêtes de transport sont issues du pavillon d'urgence. Toutefois, en termes de répartition des véhicules entre les différents pavillons, cela réduit la couverture des autres pavillons et concentre sur la couverture d'une zone limitée du campus.

Contrairement à la deuxième, la troisième stratégie, qui relocalise les véhicules au rez-de-chaussée de pavillon de livraison, vise à augmenter la répartition des véhicules sur le campus de sorte que les véhicules restent plus proches des différentes sources de demandes.

3.3.4 L'heuristique de réoptimisation

Notre approche dynamique est basée sur une heuristique de réoptimisation. Cette dernière est un algorithme dynamique qui ne prend pas en considération les prévisions de la demande, à savoir, l'aspect stochastique. Bien qu'une telle heuristique soit théoriquement simple, son adaptation à notre cas a requis certaines modifications.

Nous avons aussi intégré les stratégies présentées dans la section précédente. Ceci nous a donné trois versions différentes de cette heuristique dont chacune utilise l'un des algorithmes suivants :

- RH1 : algorithme qui utilise la première stratégie d'attente
- RH2 : algorithme qui utilise la deuxième stratégie de relocalisation
- RH3 : algorithme qui utilise la troisième stratégie de relocalisation

Pour ces heuristiques les décisions stratégiques sont considérées comme des mécanismes automatiques que nous imposons dans le but de mieux réagir à l'arrivée d'une nouvelle demande. Du coup, les décisions stratégiques ne sont pas évaluées. En d'autres mots, si nous sommes dans le cas où toutes les demandes sont assignées, alors la solution est fixée, elle ne change jamais même si des décisions de relocalisation sont exécutées. Ce qui signifie que la solution ne change que lorsqu'une nouvelle requête est révélée. Alors, nous pouvons dire que le seul événement à considérer dans notre heuristique c'est l'arrivée d'une nouvelle requête de transport.

Un aperçu général de notre heuristique de réoptimisation est donné dans l'Algorithme 3 ci-dessous. D'abord, l'algorithme crée un plan d'itinéraire vide. Ensuite, une opération d'insertion de toutes les demandes connues au début de la journée est exécutée afin de créer une solution initiale s . Puis, l'algorithme commence à générer les itinéraires pour les chauffeurs en exécutant l'algorithme de l'ALNS. La solution obtenue avec l'ALNS ne permet pas à un véhicule de sortir que s'il peut arriver à son lieu de cueillette au moins à l'heure de début de service

a_r , sinon le véhicule doit attendre à son emplacement actuel. À chaque nouvel événement qui se produit, toutes les décisions qui ont été effectuées avant l'instant t et toutes les décisions qui sont en train d'être exécutées sont verrouillées et fixées de façon permanente dans la solution s . Si un véhicule est en déplacement vers un nœud, ce dernier est verrouillé définitivement dans l'itinéraire de ce véhicule. À chaque fois qu'une nouvelle requête de transport est révélée, elle est ajoutée à s et une étape de réoptimisation avec l'ALNS est exécutée. Toutes les demandes non traitées et toutes les nouvelles demandes révélées sont insérées seulement après les nœuds fixés dans la solution ou dans un itinéraire encore vide. Lorsqu'il n'y a plus de nouvelles demandes à arriver et lorsque toutes les demandes sont intégrées dans la solution s , nous pouvons donc trouver le plan final d'itinéraires qui contient toutes les décisions et les actions de toute la journée. Dans ce travail, un plan d'itinéraires ou simplement un plan est une solution partielle pour le problème dynamique. Au fur et à mesure que le temps avance, et en fonction des nouvelles informations révélées, il peut être modifié. Le plan final représente la solution finale de l'heuristique qui intègre toutes les demandes de la journée.

Algorithme 3 : Heuristique de réoptimisation

- 1: **Entrée** : Les stratégies RH
 - 2: Construire une solution initiale s avec les demandes disponibles à $t = 0$
 - 3: Optimiser s avec l'ALNS
 - 4: **tant que** Il y a de nouvelles demandes **faire**
 - 5: Fixer les affectations passées des brancardiers dans s
 - 6: Ajouter la nouvelle demande à s
 - 7: Réoptimiser s avec l'ALNS
 - 8: Appliquer la stratégie RH sélectionnée
 - 9: **fin tant que**
-

3.4 Approche stochastique

Le but de la première et de la deuxième approche est d'étudier le problème dans les meilleures et les pires conditions possibles. En revanche, l'objectif de l'approche stochastique est d'étudier le problème dans des conditions réelles. L'idée de base de cette approche est d'utiliser les connaissances du passé pour prédire ce qui pourrait se produire dans un avenir proche. En se basant sur ces prédictions, nous pouvons développer des stratégies pour nous aider à prendre des décisions qui optimisent les flux de transport. Cependant, la quantité des données historiques à utiliser a un impact sur les résultats. Plus il y a des données historiques pris en considération dans les algorithmes, plus le temps et l'effort de calcul des algorithmes augmentent.

3.4.1 Procédure de résolution

La première partie de cette procédure est similaire à celle de la procédure dynamique. Lors de son lancement, la simulation commence par générer le scénario réel qui contient toutes les

demandes disponibles jusqu'à l'instant actuel t de l'algorithme. Ensuite, elle exécute une étape d'optimisation pour construire les itinéraires. Ce qui est nouveau dans cette approche, c'est qu'après chaque étape d'optimisation, l'algorithme génère un ensemble de scénarios contenant des demandes fictives, obtenues à l'aide d'une opération d'échantillonnage de la distribution de probabilité de leur apparition. Ces scénarios contiennent des demandes potentielles qui peuvent arriver dans le futur proche. L'algorithme de l'ALNS est exécuté pour résoudre ces scénarios afin de construire les itinéraires qui seront une entrée de la fonction de consensus. Le but de cette fonction est de choisir un emplacement pour relocaliser les véhicules.

3.4.2 L'approche de planification basée sur des scénarios

L'approche stochastique est basée sur l'approche de planification utilisant des scénarios (SBPA) fut proposé par Bent et Van Hentenryck (2004). À chaque fois qu'un événement se produit, la SBPA génère des scénarios fictifs qui peuvent se produire dans un futur proche. Ces scénarios sont obtenus par un échantillonnage des demandes réelles du passé à partir d'un historique. Afin de résoudre chaque scénario, l'algorithme de l'ALNS est exécuté. Ensuite, en utilisant une fonction de consensus, l'approche sélectionne pour chaque véhicule disponible un emplacement pour le relocaliser. L'Algorithme 4 décrit le pseudo-code de l'heuristique de planification basée sur des scénarios échantillonnés.

Les premières étapes sont les mêmes étapes que l'algorithme de réoptimisation. À l'étape 8, l'algorithme appelle la fonction "Relocalisation" qui retourne vrai ou faux. Cette fonction prend le temps courant t de l'algorithme en paramètre et ensuite elle vérifie s'il y aura des décisions dans le futur proche ($t + \Delta$). Si la fonction trouve au moins une décision dans l'intervalle de temps Δ , elle retourne vrai. En conséquence, l'algorithme génère un ensemble de scénarios Ω de demandes fictives qui peuvent se produire entre l'instant t et l'instant $t + \Delta$. Pour chaque scénario ω , l'heuristique génère un plan s_ω qui contient toutes les demandes fictives du scénario ω . Ce plan s_ω est optimisé en faisant recours à l'ALNS. Ensuite, en utilisant la fonction de consensus, chaque véhicule est relocalisé au nœud qui a le score le plus élevé.

Si la fonction "Relocalisation" ne trouve aucune décision d'affectation dans le future proche, alors la stratégie SH sélectionnée au début est appliquée. Les stratégies SH sont les mêmes stratégies utilisées dans l'heuristique de réoptimisation :

- SH1 : algorithme qui utilise la première stratégie d'attente
- SH2 : algorithme qui utilise la deuxième stratégie de relocalisation
- SH3 : algorithme qui utilise la troisième stratégie de relocalisation

La fonction de consensus prend en entrée l'ensemble de tous les itinéraires créés pour tous les scénarios. En retour, la fonction calcule pour chaque couple de véhicule-nœud un score

et relocalise chaque véhicule vers le nœud qui a le score le plus élevé. Le score est calculé en comptant pour chaque véhicule le nombre de fois que chaque nœud apparaît dans tous les plans. Le but est de relocaliser chaque véhicule vers l'endroit qui a la probabilité la plus élevée d'être sa prochaine destination. Ainsi, cette relocalisation peut être considérée comme une troisième stratégie de relocalisation.

Algorithme 4 : L'heuristique de planification basée sur des scénarios

```

1: Entrée : Les stratégies SH
2: Construire une solution  $s$  qui contient les requêtes de transport connues à  $t = 0$ 
3: Optimiser avec l'ALNS
4: tant que Il y a de nouvelles demandes faire
5:   Verrouiller toutes les décisions effectuées dans la solution  $s$ 
6:   Ajouter les nouvelles demandes à  $s$ 
7:   Réoptimiser  $s$  avec l'ALNS
8:   si Relocalisation( $t$ ) alors
9:     Générer un ensemble de scénarios  $\Omega$  de demandes fictives
10:    pour chaque scénario  $\omega$  dans  $\Omega$  faire
11:      Construire une solution  $s_\omega$  qui contient toutes les requêtes du scénario  $\omega$ 
12:      Optimiser  $s_\omega$  avec l'ALNS
13:    fin pour
14:    pour chaque véhicule  $k$  libre faire
15:      Relocaliser  $k$  vers le nœud qui a le score le plus élevé de la fonction de consensus
16:    fin pour
17:    sinon Appliquer la stratégie SH sélectionnée
18:  fin si
19: fin tant que

```

Pour mieux illustrer le fonctionnement de la fonction de consensus, nous donnons l'exemple suivant avec des demandes fictives. Nous avons trois plans dont chacun contient trois routes.

Plan 1 :

- Véhicule 1 : (36-37-36-8).
- Véhicule 2 : (36-18-37-20).
- Véhicule 3 : (37-38-37-17).

Plan 2 :

- Véhicule 1 :(36-16).
- Véhicule 2 : (161-162-165-164).

Plan 3 :

- Véhicule 1 : (41-40-16-19).
- Véhicule 2 : (161-166).
- Véhicule 3 : (18-40-21-166).

La fonction de consensus calculera l'occurrence des nœuds pour chaque véhicule et elle re-

tourne le nœud le plus commun. Dans cet exemple elle donne les résultats suivants. Véhicule 1 doit se relocaliser vers le nœud 36, le véhicule 2 vers le nœud 161 et le véhicule 3 vers le nœud 37. Dans le cas où deux nœuds ont le même score, l'algorithme choisi le premier qui se trouve dans la liste des scores.

Chapitre 4

Étude expérimentale

Introduction

Ce chapitre est dédié à la présentation des résultats des différentes approches développées dans ce mémoire. Dans la première partie, nous présentons les instances tests utilisées. La deuxième partie décrit la démarche utilisée pour choisir les valeurs de certains paramètres. Ensuite, la troisième partie présente les résultats obtenus par chaque approche suivie d'une comparaison. La quatrième partie offre une analyse de sensibilité des résultats dû au changement du nombre de véhicules disponibles.

4.1 Conception des instances tests

Les données utilisées dans ce travail pour élaborer les instances sont les données réelles collectées de l'hôpital universitaire Policlinico Sant'Orsola-Malpighi. Ces données contiennent toutes les informations sur les requêtes de transport reçues par les brancardiers au cours de l'année 2016 et ont été utilisées pour tester et évaluer les différentes approches développées.

Dans le cadre de ce travail, nous avons des instances réelles que nous appelons des "scénarios réels" et d'autres instances stochastiques que nous appelons simplement des "scénarios". Un générateur a été développé pour créer ces instances. Ce dernier génère un fichier texte contenant toutes les requêtes de transport qui sont connues au début de la journée ainsi que celles qui sont révélées au cours de la journée. Le fichier est ordonné selon le temps d'arrivée des demandes.

Pour les instances réelles, la procédure de création consiste à prendre toutes les informations réelles relatives aux demandes de transport de chaque jour et à générer des instances lisibles par l'algorithme pour représenter la situation des flux de transport réels au sein de l'hôpital. Un total de 365 fichiers texte ont été créés où chacun représente un scénario réel contenant toutes les requêtes de transport réalisées au cours de l'année 2016.

En ce qui concerne les instances stochastiques, une procédure de création différente a été utilisée. Le but de ces instances est de représenter des scénarios possibles pour une journée de travail donnée et de les utiliser dans l'approche stochastique afin de trouver des relocalisations pour les véhicules. Pour cela, nous avons créé des instances stochastiques pour chaque jour de la semaine en nous servant de l'étude statistique de l'évolution du nombre de requêtes de transport journalier. La tendance de la demande journalière est représentée dans la Figure 2.4. Nous pouvons constater que la demande devient atypique et baisse énormément durant certains jours. Dans le domaine des statistiques, ce phénomène est appelé bruit et nécessite des corrections à faire. Alors, nous avons procédé à une correction des données en éliminant les nombres de demandes qui se situe hors de la plage de 15% de la moyenne pour chaque jour. Ensuite, nous avons examiné la distribution de densité des demandes pour les sept jours de la semaine. Comme dans les études similaires à notre travail, nous avons mis l'hypothèse que l'évolution du nombre de requêtes de transport est un phénomène aléatoire qui suit la loi normale. Donc, pour chaque jour i , le générateur calcule les paramètres de cette loi qui consiste à sa moyenne μ_i , son écart type σ_i et sa variance σ_i^2 , pour chaque $i \in \{1, 2, \dots, 365\}$. Par la suite, l'algorithme prend un nombre x_i aléatoire de la distribution suivante :

$$f(x_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}}.$$

Subséquentement, il lance une opération d'échantillonnage aléatoire, pour les x_i requêtes de transport, à partir de l'ensemble des données réelles du jour i de la semaine.

4.2 Paramètres

L'objectif de cette partie est d'identifier des valeurs intéressantes pour le nombre d'itérations η de l'ALNS et le nombre de scénarios à utiliser dans l'approche stochastique. Pour ce faire, nous avons conçu un ensemble de 35 instances tests composées de cinq instances pour chaque jour de la semaine. Ces dernières sont générées aléatoirement de la même façon que nous avons généré les instances stochastiques.

4.2.1 Nombre d'itérations de l'ALNS

Cette sous-section présente des tests avec différents nombres d'itérations d'ALNS. L'objectif est de trouver la bonne valeur pour ce paramètre. Le Tableau 4.1 représente les résultats de ces tests. La première colonne indique l'algorithme utilisé dans les tests. Les trois versions de l'heuristique de réoptimisation sont représentées par "RH1", "RH2" et "RH3" utilisant respectivement la première, la deuxième et la troisième stratégie. Dans tous les tests, tous les paramètres sont les mêmes, sauf le nombre d'itérations qui change. Les critères d'évaluation

considérés sont le retard en minutes, la distance en kilomètre et le temps d'exécution en secondes. La dernière ligne du tableau indique la moyenne pour chaque colonne.

Tableau 4.1 – Tester différentes itérations de l'ALNS

Algorithmes	Itérations de l'ALNS = 100			Itérations de l'ALNS = 250			Itérations de l'ALNS = 500			Itérations de l'ALNS = 1000			Itérations de l'ALNS = 2000		
	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps
Statique	10.1	1,619.1	0.6	10.1	1,613.0	0.7	10.1	1,610.7	1.1	10.0	1,609.1	1.6	10.0	1,607.2	2.9
RH1	11.4	1,640.6	0.5	11.3	1,630.3	0.5	11.4	1,630.0	0.5	11.4	1,629.5	0.6	11.2	1,630.4	0.7
RH2	11.5	1,640.5	0.5	11.5	1,639.8	0.5	11.4	1,639.9	0.5	11.3	1,639.1	0.7	11.3	1,639.1	0.7
RH3	11.3	1,632.4	0.5	11.1	1,631.9	0.5	11.3	1,632.4	0.5	10.9	1,631.5	0.6	11.4	1,630.3	0.7
Moyenne	11.1	1,633.2	0.5	11.0	1,628.7	0.6	11.1	1,628.2	0.7	10.9	1,627.3	0.9	11.0	1,626.8	1.2

Les résultats du Tableau 4.1 indiquent qu'en augmentant le nombre d'itérations η la distance diminue et le temps d'exécution augmente. Cependant, l'augmentation du nombre d'itérations n'implique pas forcément une diminution du retard. Ceci peut être expliqué par les instances utilisées où il y a une forte demande et des fenêtres de temps avec courtes durées. En se basant sur ces résultats, le nombre d'itérations choisi pour les trois approches est $\eta = 500$. En fait, avec ce nombre, l'ALNS peut garantir un bon compromis entre le temps d'exécution et les résultats.

4.2.2 Nombre de scénarios

Le choix du nombre de scénarios à utiliser dans l'approche stochastique est très important. Ce choix affecte les résultats et surtout le temps d'exécution qui augmente généralement en fonction du nombre de scénarios. Le Tableau 4.2 représente les résultats de tests des différents algorithmes stochastiques en utilisant 5, 10, 15, 20, 25 et 30 scénarios. Les algorithmes stochastiques 1,2 et 3 représentent l'approche de planification basée sur des scénarios échantillonnés en utilisant respectivement la première, la deuxième et la troisième stratégie. La première colonne indique le nombre de scénarios utilisés. Pour chaque algorithme, nous donnons le retard moyen en minutes, la distance moyenne parcourue en kilomètres et le temps d'exécution moyen en secondes. La dernière colonne du tableau représente le retard moyen enregistré pour chaque nombre de scénarios avec les trois stratégies.

Tableau 4.2 – Tester différents nombres de scénario

# de scénarios	Stochastique 1			Stochastique 2			Stochastique 3			Retard moyen
	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps	
5	14.9	1661.5	3.5	14.9	1714.2	3.3	15.1	1679.0	2.9	15.0
10	13.4	1670.5	7.0	12.4	1712.4	6.5	9.7	1682.4	6.1	11.9
15	12.7	1664.6	8.9	15.3	1715.3	9.0	14.0	1670.8	7.5	14.0
20	19.9	1662.4	10.7	9.7	1705.1	12.4	12.5	1684.4	15.0	14.0
25	15.9	1660.1	14.3	16.3	1710.5	14.0	12.5	1684.4	15.0	14.9
30	18.3	1672.0	16.0	13.6	1708.4	16.6	19.5	1695.6	16.4	17.1
Moyenne	15.9	1665.2	10.1	13.7	1711.0	10.3	13.9	1682.8	10.5	14.5

L'analyse des résultats obtenus révèle que l'augmentation du nombre de scénarios utilisés

n'améliore pas forcément les résultats. Ce qui est plus important, est la qualité des scénarios échantillonnés utilisés. Plus l'ensemble de ces scénarios est similaire au scénario réel, plus les résultats sont améliorés. D'ailleurs, nous remarquons que les résultats obtenus sont proches en termes de retard et de distance. Cependant, le temps d'exécution augmente en fonction du nombre de scénarios. Les meilleurs résultats obtenus en termes de retard moyen sont ceux avec 10 scénarios.

Dans le reste de ce chapitre, le nombre de scénarios à utiliser pour chaque algorithme est égal à 10.

4.3 Présentation des résultats finaux

Cette section présente les résultats des tests de chaque approche sur les 365 instances dont chacune correspond à un jour de l'année 2016. Pour ces tests, nous considérons la flotte réelle de l'hôpital qui est composé de cinq véhicules, dont deux ambulances et trois véhicules électriques.

À l'exception du nombre d'itérations qui est fixé à 500, tous les paramètres de l'ALNS ont les mêmes valeurs que celles utilisées dans le travail de Ropke et Pisinger (2006). L'implémentation de tous les algorithmes a été réalisée en C++ et compilée avec Clang. Tous les tests ont été effectués sur un ordinateur équipé d'une puce Apple M1 dont le processeur est composé de 8 cœurs @ 3,2 GHz de fréquence.

4.3.1 Résultats de l'approche statique

Les résultats de l'approche statique sur les 365 instances réelles ont été regroupés dans le tableau 4.3. La première colonne du tableau contient le mois de l'année, la deuxième colonne indique le nombre moyen de patients, la troisième colonne présente le retard total moyen en minutes, la quatrième colonne donne la distance moyenne parcourue en kilomètres et la dernière colonne décrit le temps d'exécution en secondes. Les trois dernières lignes du tableau présentent respectivement le minimum, le maximum et la moyenne trouvés dans les 365 jours. En simulant sur l'ensemble de l'année, l'approche statique a engendré un retard total moyen de 0.8 minute pour une distance moyenne parcourue de 1,341.7 kilomètres et dans un temps d'exécution moyen de 5.7 secondes. Les résultats trouvés par cette approche représentent une borne inférieure que les autres approches essayent d'atteindre, mais dans des conditions plus réelles.

4.3.2 Résultats de l'approche dynamique

Cette section présente les résultats de l'heuristique de réoptimisation. Les résultats des tests sont indiqués dans le Tableau 4.4. Pour chaque mois de l'année, ce tableau indique le nombre moyen de patients, qui reste le même pour toutes les approches, le retard moyen en minutes

Tableau 4.3 – Résultats de l’approche statique

Mois	Patients	Retard	Distance	Temps
Janvier	375.8	0.4	1,222.7	2.5
Février	438.3	0.6	1,406.4	3.8
Mars	445.2	2.0	1,491.2	5.3
Avril	462.5	0.4	1,440.6	7.7
Mai	443.5	0.0	1,360.5	6.4
Juin	436.8	2.9	1,374.8	6.0
Juillet	410.2	0.3	1,326.6	7.0
Août	336.6	0.3	957.5	2.7
Septembre	435.3	0.8	1,325.5	6.5
Octobre	431.5	0.7	1,408.2	6.6
Novembre	445.2	0.6	1,376.0	6.2
Décembre	451.6	0.5	1,416.9	6.9
Moyenne	425.7	0.8	1,341.7	5.7
Min	106.0	-	338.5	0.0
Max	663.0	64.3	2,231.5	40.0

et la distance moyenne parcourue en kilomètre pour chaque version de l’algorithme. Les algorithmes dynamique 1, dynamique 2 et dynamique 3 sont les trois versions de l’heuristique de réoptimisation en utilisant les trois différents algorithmes RH1, RH2 et RH3.

L’utilisation du premier algorithme RH1, où les véhicules sont obligés d’attendre les demandes dans leurs dernier lieu de livraison, a engendré un retard total de 16.9 minutes et une distance parcourue de 1,531.41 km, soit une augmentation d’environ 14% par rapport au cas statique. Ces résultats ont été améliorés en utilisant le deuxième algorithme qui emploie la deuxième stratégie de relocalisation. Cette deuxième stratégie consiste à faire retourner tous les véhicules au dépôt central situé aux urgences après la dernière opération livraison. Le retard moyen a diminué avec plus de 19% pour atteindre une moyenne de 14.3 minutes. Toutefois, la distance parcourue a augmenté de 6% par rapport aux résultats de RH1 et d’environ 21% par rapport aux résultats de l’approche statique. Ceci peut être expliqué par le retour au pavillon de l’urgence qui est imposé par la deuxième stratégie. En revanche, le troisième algorithme dynamique, qui intègre la troisième stratégie, était capable de trouver de meilleurs résultats en termes de retard que les deux algorithmes précédents. Cette dernière propose que tous les véhicules doivent être relocalisés au rez-de-chaussée du dernier pavillon où les brancardiers ont effectué leur dernière livraison et d’attendre les prochaines affectations à cet endroit. En fait, le retard et la distance ont été allégés à 13.88 minutes et 1,547.4 km.

En termes de temps d’exécution, les trois algorithmes dynamiques ont été plus rapides par rapport à l’algorithme statique. L’heuristique de réoptimisation avec ces trois algorithmes RH1, RH2 et RH3 a réussi à réduire le temps d’exécution à moins d’une seconde. Ceci peut

être expliqué par le fait que dans l’approche statique, l’algorithme traite toutes les demandes au début. Cependant, dans l’approche dynamique, les demandes sont traitées selon leurs temps de disponibilité. Donc à chaque itération, lorsque les décisions passées sont verrouillées, l’algorithme traite les demandes disponibles à l’instant t uniquement.

Tableau 4.4 – Résultats de l’approche dynamique

Mois	Patients	Dynamique 1		Dynamique 2		Dynamique 3	
		Retard	Distance	Retard	Distance	Retard	Distance
Janvier	375.8	10.1	1,372.4	9.8	1,454.9	9.3	1,411.3
Février	438.3	18.7	1,583.1	13.3	1,669.9	14.2	1,606.8
Mars	445.2	33.1	1,671.0	27.2	1,747.2	24.0	1,674.8
Avril	462.5	14.3	1,612.4	12.2	1,731.3	10.8	1,640.1
Mai	443.5	16.3	1,545.5	11.7	1,629.8	11.9	1,549.5
Juin	436.8	14.0	1,557.1	14.7	1,657.9	18.1	1,591.3
Juillet	410.2	15.2	1,536.9	10.3	1,635.1	9.5	1,545.9
Août	336.6	3.7	1,170.5	3.0	1,211.7	3.1	1,144.4
Septembre	435.3	10.3	1,525.3	9.2	1,631.5	9.3	1,547.7
Octobre	431.5	12.8	1,598.5	14.1	1,707.0	12.8	1,626.5
Novembre	445.2	37.1	1,599.0	34.7	1,710.7	32.1	1,624.6
Décembre	451.6	18.5	1,599.4	12.4	1,707.4	12.9	1,602.7
Moyenne	425.7	16.9	1,531.4	14.3	1,625.2	13.9	1,547.4
Min	106.0	-	357.0	-	358.7	-	362.9
Max	663.0	281.5	2,384.2	241.1	2,531.8	231.3	2,457.1

Les résultats trouvés par l’approche dynamique montrent que les algorithmes RH2 et RH3 donnent de meilleurs résultats que l’algorithme RH1. Ce qui souligne l’importance de la relocalisation des véhicules en général ainsi que son effet sur les résultats en l’intégrant avec l’heuristique de réoptimisation.

Bien que ces résultats montrent que la troisième stratégie fournit des solutions avec moins de retard et moins de distance parcourue sur toute l’année, la deuxième stratégie donne de meilleurs résultats en termes de retard, particulièrement en février, mai, juin, août, septembre et décembre. Ceci est lié aux phénomènes aléatoires qui caractérisent un tel contexte réel de travail. En fait, le trafic des flux de patients est causé soit en raison des déplacements urgents, qui émergent du pavillon d’urgence, soit en raison des déplacements prévus ou non urgents qui émergent des autres pavillons. Étant donné que le dépôt est situé au même pavillon des services d’urgences, la deuxième stratégie qui relocalise les brancardiers au dépôt donne de meilleurs résultats durant les jours où les demandes de transport planifiées et non urgentes sont faibles.

Pour mieux comprendre la performance de ces deux stratégies, nous avons analysé chaque relocalisation effectuée par eux. Une fonction est développée afin de calculer, pour chaque relocalisation effectuée, le temps gagné ou le temps perdu. Si une relocalisation engendre une

perte du temps, elle est appelée une "mauvaise relocalisation", et si elle entraîne un gain du temps, elle est appelée une "bonne relocalisation". Afin de mieux comprendre comment chaque relocalisation est évaluée, nous fournissons l'exemple suivant.

Soit k un véhicule donné, i le dernier nœud de livraison pour ce véhicule, r le nœud où k a été relocalisé, p le prochain nœud de cueillette et t_{xy} est le temps de déplacement entre deux nœuds x et y . Alors, le temps perdu T_p et le temps gagné T_g sont calculés comme suivant :

$$\left\{ \begin{array}{ll} \textit{si} & t_{rp} - t_{ip} > 0 \\ \textit{alors} & T_g = t_{rp} - t_{ip} \\ \textit{sinon} & \textit{si} \quad t_{rp} - t_{ip} \leq 0 \\ \textit{alors} & T_p = t_{ip} - t_{rp} \end{array} \right.$$

Les Tableaux 4.5 et 4.6 montrent les résultats d'évaluation de la deuxième et la troisième stratégie. La première colonne indique le mois de l'année, la deuxième contient le nombre de patients moyen, la troisième fournit le nombre de relocalisations total, la quatrième donne le nombre de mauvaises relocalisations, la cinquième présente le temps perdu, la sixième colonne indique les bonnes relocalisations et la dernière colonne contient le temps gagné. Contrairement aux résultats numériques des deux algorithmes RH1 et RH2, ces deux tableaux montrent que la stratégie 2 est plus efficace que la troisième stratégie. En fait, la deuxième stratégie a effectué moins de 49.36% de mauvaises relocalisation, ce qui a sauvé environ 30% du temps perdu. Bien qu'elle fasse moins de 11.66% de bonnes relocalisation que la troisième stratégie, elle a réalisé environ plus de 22% du temps gagné.

Bien que la deuxième stratégie effectue moins de mauvaises relocalisations, moins du temps perdu et gagne plus du temps en relocalisant les véhicules au dépôt, la troisième stratégie donne de meilleurs résultats en la combinant avec l'heuristique de réoptimisation. Ceci peut être expliqué par plusieurs facteurs. En fait, dans le cas étudié, plus de 51% des requêtes de transport émergent des services d'urgence, donc, les décisions de relocalisation vers le pavillon d'urgence rapprochent plus les véhicules des sources potentielles de demandes de transport et donc aide à gagner plus de temps. Cependant, l'utilisation de cette dernière dans notre problème, où les fenêtres de temps sont trop serrées, implique généralement des retards pour les livraisons vers des pavillons loin de l'urgence. En revanche, la troisième stratégie est capable de compenser ces points faibles vu qu'elle répartit les véhicules de manière plus équilibrée sur le campus hospitalier. Avec cette dernière stratégie, il y aura toujours des véhicules proches de pavillon de l'urgence et en même temps d'autres pavillons, ce qui minimise plus les retards et les distances parcourues avec cette stratégie.

4.3.3 Résultats de l'approche stochastique

L'approche stochastique développée dans ce mémoire est fondée sur une approche de planification basée sur des scénarios échantillonnés. Comme dans l'approche dynamique, cette

Tableau 4.5 – Analyse de la deuxième stratégie

Mois	Patients	# de relocalisations	Mauvaises relocalisations	Temps perdu	Bonnes relocalisations	Temps Gagné
Janvier	375.8	39	5	15	34	131
Février	438.3	39	5	15	34	129
Mars	445.2	38	5	18	32	131
Avril	462.5	39	6	16	33	130
Mai	443.5	42	7	21	35	137
Juin	436.8	38	6	21	32	128
Juillet	410.2	37	5	20	32	128
Août	336.6	39	7	24	32	112
Septembre	435.3	36	5	15	31	118
Octobre	431.5	39	5	14	33	134
Novembre	445.2	39	5	17	34	138
Décembre	451.6	41	5	15	36	139
Moyenne	425.7	39	5.7	17.5	33.1	129.5
Min	106.0	15	-	-	15	45
Max	663.0	62	14	76	48	245
Somme	5,112.4	14,160	2,069	6,393	12,091	47,284

Tableau 4.6 – Analyse de la troisième stratégie

Mois	Patients	# de relocalisations	Mauvaises relocalisations	Temps perdu	Bonnes relocalisations	Temps Gagné
Janvier	375.8	39	9	18	37	107
Février	438.3	39	10	20	35	98
Mars	445.2	38	9	19	38	110
Avril	462.5	39	11	24	39	110
Mai	443.5	42	11	24	40	112
Juin	436.8	38	12	27	37	107
Juillet	410.2	37	11	28	36	104
Août	336.6	39	14	31	34	95
Septembre	435.3	36	11	26	37	104
Octobre	431.5	39	12	30	37	102
Novembre	445.2	39	11	27	40	112
Décembre	451.6	41	13	30	41	118
Moyenne	425.7	49	11.2	25.2	37.5	106.6
Min	106.0	21	-	-	21	53
Max	663.0	96	26	66	70	193
Somme	5,112.4	17,773	4,086	9,216	13,687	38,891

approche utilise également les trois stratégies. Donc, nous avons également obtenu trois versions différentes de cette heuristique. En plus, d'utiliser les 365 instances réelles comme les approches précédentes, cette approche utilise également 10 scénarios stochastiques. Le Tableau 4.7 représente les résultats des trois versions de l'heuristique, stochastique 1, stochastique 2 et stochastique 3 qui utilisent respectivement les algorithmes SH1, SH2 et SH3.

Les résultats montrent que l'intégration des prévisions des demandes a amélioré la performance de l'algorithme. En fait, une comparaison des résultats du premier algorithme stochastique avec ceux du premier algorithme dynamique indique que le retard total moyen a diminué d'environ 25% pour donner un total de 12.7 minutes. Toutefois, la distance moyenne parcourue a eu une hausse de 3.7%. Également, l'utilisation de l'algorithme SH2 a réussi à baisser le retard jusqu'à 10.3 minutes, ce qui représente une amélioration d'environ 28% par rapport à l'algorithme RH2. Cependant, la distance moyenne parcourue a augmenté de 15.2 km, soit 0.9%. Avec le troisième algorithme SH3, le retard total moyen est de 10.4 minutes, soit 25% moins que le troisième algorithme dynamique qui utilise le RH3. En contrepartie, la distance parcourue a augmenté d'environ 1.5%.

Tableau 4.7 – Résultats de l'approche stochastique

Mois	Patients	Stochastique 1		Stochastique 2		Stochastique 3	
		Retard	Distance	Retard	Distance	Retard	Distance
Janvier	375.8	12.0	1,433.8	9.6	1,473.6	8.5	1,433.7
Février	438.3	15.2	1,642.8	11.1	1,683.5	12.1	1,642.5
Mars	445.2	27.5	1,708.3	18.0	1,754.3	24.6	1,726.3
Avril	462.5	12.4	1,692.4	11.7	1,740.3	11.8	1,705.5
Mai	443.5	9.4	1,595.9	10.1	1,652.5	9.1	1,601.5
Juin	436.8	16.2	1,645.1	11.7	1,700.5	12.1	1,669.2
Juillet	410.2	8.2	1,598.4	8.3	1,668.9	7.1	1,616.7
Août	336.6	3.5	1,202.4	3.8	1,232.4	2.9	1,200.0
Septembre	435.3	7.5	1,587.4	6.1	1,652.6	8.7	1,606.8
Octobre	431.5	9.2	1,652.8	8.9	1,699.1	6.8	1,661.4
Novembre	445.2	14.3	1,622.5	13.2	1,694.0	11.0	1,645.1
Décembre	451.6	17.1	1,672.6	9.8	1,735.8	10.2	1,676.1
Moyenne	425.7	12.7	1,587.8	10.3	1,640.4	10.4	1,598.6
Min	106.0	-	357.0	-	358.7	-	362.9
Max	663.0	155.2	2,377.9	140.7	2,518.7	167.3	2,424.1

Étant donné que notre objectif principal est de minimiser le retard total pondéré, nous pouvons déduire que les algorithmes stochastiques fournissent de meilleurs résultats que les algorithmes dynamiques. Les résultats marquent que les algorithmes stochastiques engendrent toujours moins de retard que les algorithmes dynamiques. Cependant, il faut mentionner qu'avec l'approche stochastique, le temps d'exécution a augmenté à une moyenne de 5.95 secondes et ceci à cause du nombre de scénarios utilisés.

La deuxième stratégie, qui consiste à relocaliser les véhicules à l’urgence, donne de meilleurs résultats. Cependant, la troisième stratégie est capable de trouver des solutions proches de la deuxième. En effet, selon le Tableau 4.7, il y a seulement une différence de 0.1 minute de retard.

4.3.4 Discussion et interprétation des résultats

Une comparaison entre les résultats des différents algorithmes montre qu’en termes de minimisation des retards, l’approche de planification basée sur des scénarios échantillonnés a une performance plus élevée que l’heuristique de réoptimisation. En revanche, cette dernière heuristique donne toujours des solutions avec des distances plus courtes et un temps d’exécution plus rapide que les autres algorithmes. Avec ses trois versions différentes, cette heuristique est capable de trouver une bonne solution en moins d’une seconde. Dans un contexte de planification réel, le temps d’exécution est très important puisque le planificateur doit prendre des décisions en temps réel.

En ce qui concerne les stratégies, nous remarquons que la troisième stratégie donne les meilleurs résultats avec l’heuristique de réoptimisation. Néanmoins, la deuxième stratégie est capable de donner de meilleures performances avec l’approche stochastique en termes de retard. En fait, l’approche de planification basée sur des scénarios échantillonnés est capable de corriger le point faible de la deuxième stratégie en utilisant la fonction de consensus qui relocalise les véhicules en se basant sur des prévisions. Si la fonction de consensus sélectionne un lieu pour la relocalisation, le véhicule se déplace vers ce lieu, sinon, il se déplace vers le dépôt situé à l’urgence. L’utilisation de cette dernière stratégie avec la fonction de consensus permet de garder certains véhicules à l’urgence et d’avoir d’autres proches des différentes sources potentielles de demandes.

Le Tableau 4.8 résume les résultats de tous les algorithmes. La première colonne contient le nom de l’algorithme utilisé, la deuxième, le retard total moyen, la troisième indique la distance moyenne parcourue, et la quatrième présente le temps d’exécution moyen. Tous les tests sont effectués sur les 365 instances et les algorithmes stochastiques utilisent 10 scénarios.

Tableau 4.8 – Les résultats des différents algorithmes

Algorithme	Retard	Distance	Temps
Statique	0.8	1341.7	5.6
RH1	16.9	1531.4	0.4
RH2	14.3	1625.2	0.5
RH3	13.9	1547.4	0.4
SH1	12.7	1587.8	5.8
SH2	10.3	1640.4	6.2
SH3	10.4	1598.6	5.8
Moyenne	11.3	1,553.2	3.5

Il faut mentionner que même si les résultats des instances réelles ont montré qu'il y a des algorithmes capables de trouver de meilleures solutions que d'autres, tous les algorithmes restent efficaces et capables de trouver de bons résultats.

Le Tableau 4.8 montre qu'il n'y a pas une grande différence entre le retard engendré par les algorithmes dynamiques et stochastiques. En fait, le plus grand retard, 16.9 minutes, est trouvé en utilisant le premier algorithme dynamique, tandis que le plus petit retard, 10.3 minutes, est trouvé en utilisant le deuxième algorithme stochastique. C'est une différence de moyenne 6.6 minutes par jour. Cette différence est négligeable si nous prenons en considération que c'est un retard accumulé sur toute une journée de travail en livrant plus de 425 patients en moyenne.

Pour cela nous avons décidé d'effectuer un test statistique qui nous permet de comparer significativement les résultats de SH2 avec les autres algorithmes. Afin de choisir le test convenable, nous avons fait un test de normalité de Shapiro-Wilk à l'aide du logiciel "R". Le niveau de signification est $\alpha = 5\%$. Le Tableau 4.9 montre les valeurs des statistiques du test de Shapiro-Wilk. Dans ce test, si les $p - values$ sont supérieurs au niveau de signification α l'échantillon de données suit une loi normale. Sinon, l'échantillon de données ne suit pas une loi normale.

Tableau 4.9 – Les statistiques du test de normalité de Shapiro-Wilk

	Statique	RH1	RH2	RH3	SH1	SH2	SH3
V	0.18879	0.60989	0.51731	0.51155	0.59876	0.64711	0.58242
$p - values$	$2.2e^{-16}$	$2.9e^{-16}$	$2.2e^{-16}$	$2.2e^{-16}$	$3.7e^{-16}$	$2.5e^{-16}$	$2.2e^{-16}$

NB : Les colonnes sont colorées en gris lorsque la normalité n'est pas acceptée.

D'après le Tableau 4.9, nous pouvons remarquer que aucun échantillon de données suit une loi normale. Ceci peut être aussi confirmé grâce aux histogrammes de retard de chaque chaque approche présentée dans la Figure 4.1.

Enfin, les retards trouvés dans les solutions sont analysés en effectuant un test unilatéral de Wilcoxon (Wilcoxon, 1992). Ce dernier est un test non paramétrique qui permet de comparer deux groupes indépendants et il est réalisé par le logiciel *R*. La seule hypothèse requise pour réaliser ce test est que les observations sont indépendantes. Dans ce qui suit, nous considérons l'hypothèse nulle H_0 et l'hypothèse alternative H_a , où X représente l'algorithme statique ou RH1 ou RH2 ou RH3 ou SH1 ou SH3 :

- $H_0 : Moyenne(SH2) = Moyenne(X)$.

- $H_a : Moyenne(SH2) > Moyenne(X)$.

Dans ce test, le retard moyen trouvé par le deuxième algorithme stochastique SH2 est comparé avec les résultats des autres algorithmes. L'hypothèse nulle H_0 suppose que le retard obtenu

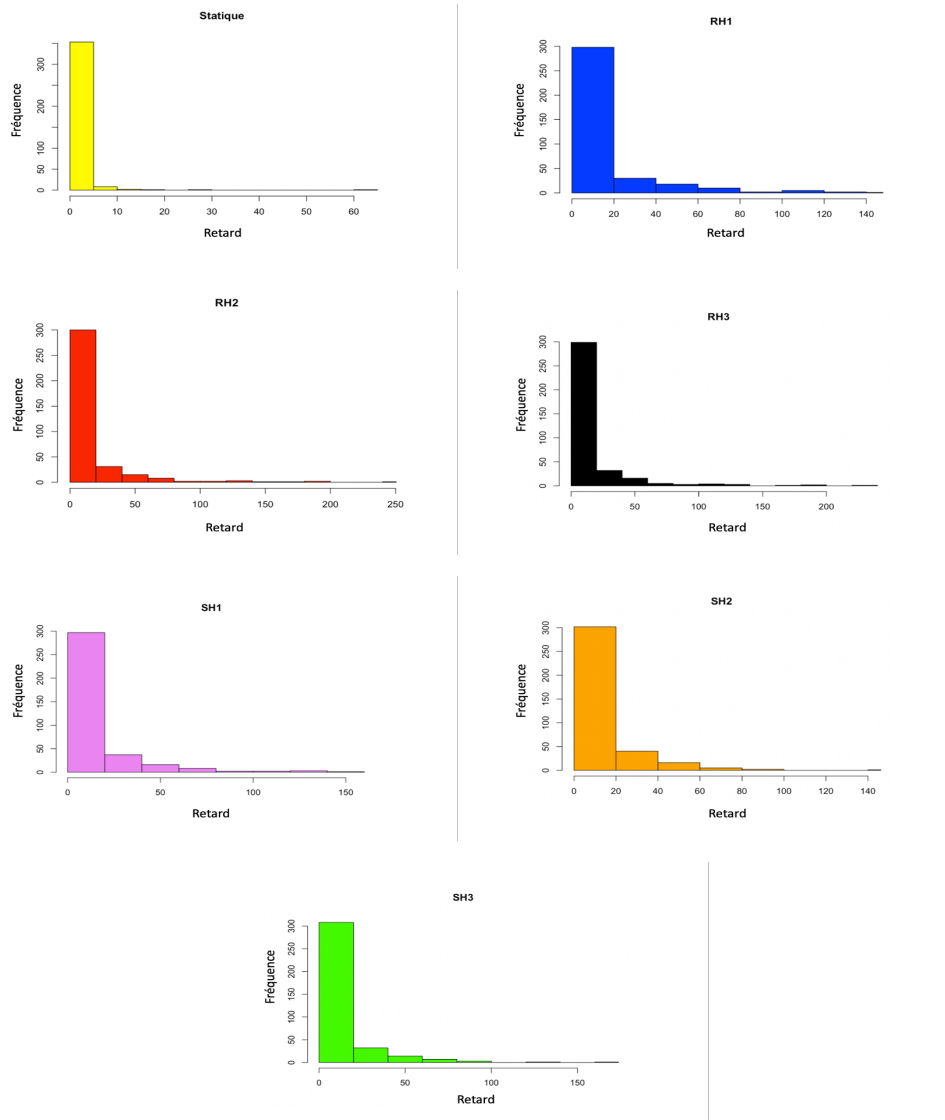


FIGURE 4.1 – Histogramme des retards pour chaque approche

en utilisant l’algorithme SH2 est égal à la valeur du retard obtenu par l’algorithme alternatif X . Le rejet de H_0 signifie que le retard moyen obtenu par SH2 est inférieur à celui trouvé par l’algorithme X . La taille de l’échantillon de données est 12. Le niveau de signification global α est fixé à 5%. Nous avons effectué plusieurs comparaisons, ce qui a nécessité d’appliquer la correction de Bonferroni (Dunn, 1961) afin de compenser le taux d’erreur par famille. La correction est faite aussi à l’aide de l’outil R . Si $p - value < \alpha = 0.05$ alors l’hypothèse correspondante doit être rejetée. Les valeurs de $p - value$ obtenues sont présentées dans le Tableau 4.10.

Dans la comparaison avec le cas statique, l’hypothèse nulle H_0 doit être rejetée puisque la valeur de $p - value < \alpha$. Cependant, il n’y a pas de raison pour rejeter l’hypothèse alternative H_a puisque la valeur de $p - value = 1$ est supérieure à $\alpha = 0.05$. Ceci signifie que statisti-

Tableau 4.10 – Les valeurs des statistiques

	Statique	RH1	RH2	RH3	SH1	SH3
$p - values(H_0)$	0.000488	0.000976	0.002441	0.010760	0.012210	0.753600
$p - values(H_a)$	1.000000	0.000488	0.00122	0.005379	0.006104	0.652600
Mediane	0.000000	4.3663000	4.321000	3.957000	4.562020	3.534000
Écart-type	4.052525	23.071440	28.773500	28.541320	22.072620	18.777890

NB : Les cellules sont colorées en gris lorsque l'hypothèse correspondante est rejetée.

quement l'algorithme statique est capable de résoudre le problème avec moins de retard que l'algorithme SH2. Pour les comparaisons, avec RH1, RH2, RH3 et SH1 l'hypothèse nulle H_0 et l'hypothèse alternative H_a doivent être rejetée puisque toutes les valeurs de $p - values$ sont inférieurs au niveau de signification α . Ce qui signifie que statiquement l'algorithme SH2 engendre moins de retard total pondéré que les algorithmes RH1, RH2, RH3 et SH1. La comparaison statistique entre SH2 et SH3 a donné des $p - values$ plus grand que le niveau de signification α , ce qui mène a accepté les deux hypothèses H_0 et H_a . Cela signifie que statistiquement le retard engendré par SH2 est supérieur ou égal à celui engendré par SH3.

4.4 Analyse de sensibilité

Le but de cette section est d'étudier et d'analyser l'effet du changement du nombres de véhicules sur les résultats des différents algorithmes.

Dans les parties précédentes, le nombre de véhicules utilisés correspond au nombre de véhicules de la flotte réelle de l'hôpital. Cette dernière est composée de deux ambulances et trois véhicules électriques. Dans ce qui suit, nous allons tester l'impact de diminuer et d'augmenter un véhicule de chaque type. Comme dans les parties précédentes, nous avons effectué les tests sur les 365 instances réelles.

Le Tableau 4.11 récapitule les résultats d'utiliser différents nombres de véhicules avec les différents algorithmes. Le nombre de véhicules à tenter dans chaque test est décrit comme suivant :

- 1A & 3VE : une ambulance et trois véhicules électriques.
- 2A & 2VE : deux ambulances et deux véhicules électriques.
- 2A & 3VE : deux ambulances et trois véhicules électriques.
- 3A & 3VE : trois ambulances et trois véhicules électriques.
- 2A & 4VE : deux ambulances et quatre véhicules électriques.

Ces résultats obtenus montrent que la réduction d'un seul véhicule engendre beaucoup de

retards, et augmente le temps d'exécution. La diminution du nombre d'ambulances affecte plus les résultats. En effet, en réduisant le nombre d'ambulances avec une unité, le retard a augmenté de 11.3 minutes jusqu'à 146.4 minutes. Aussi, en diminuant le nombre de véhicules électriques avec une unité, le retard a augmenté jusqu'à atteindre une moyenne de 49.6 minutes.

En revanche, une augmentation d'une seule ambulance peut réduire les retards jusqu'à un total de 2 minutes et une augmentation d'un seul véhicule électrique diminue les retards jusqu'à 4.6 minutes.

De plus, nous remarquons que, quel que soit le type de véhicule à ajouter, la deuxième stratégie fournit de meilleurs résultats avec l'heuristique de réoptimisation. Toutefois, la troisième stratégie donne de meilleurs résultats avec l'approche de planification basée sur des scénarios échantillonnés. Ce qui est différent du cas où nous avons diminué le nombre de véhicules.

Dans ce dernier cas, la troisième stratégie de relocalisation apporte de meilleures performances avec l'heuristique de réoptimisation. Aussi, avec l'approche de planification basée sur des scénarios échantillonnés, cette dernière stratégie donne de meilleures solutions lorsque le nombre d'ambulances est diminué, tandis que lorsque le nombre de véhicules électriques est réduit, la stratégie de retour au dépôt a fourni de meilleures performances.

Tableau 4.11 – L'impact du changement de nombre de véhicules

Algorithmes	1A & 3EV			2A & 2EV			2A & 3EV (flotte utilisée)			3A & 3EV			2A & 4EV		
	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps	Retard	Distance	Temps
Statique	42.5	1,536.9	8.9	12.6	1,397.6	7.1	0.8	1,341.7	5.6	0.3	1,283.6	4.4	0.4	1,319.8	4.4
RH1	163.7	1,669.2	1.1	55.6	1,544.4	0.6	16.9	1,531.4	0.4	3.0	1,517.5	0.3	6.5	1,514.2	0.4
RH2	165.2	1,774.7	1.2	60.2	1,628.7	0.7	14.3	1,625.2	0.5	2.0	1,591.3	0.3	2.2	1,544.2	0.3
RH3	156.6	1,682.3	1.1	54.3	1,556.7	0.6	13.9	1,547.4	0.4	2.2	1,544.2	0.3	5.1	1,531.1	0.4
SH1	171.2	1,707.0	4.5	55.2	1,581.4	4.9	12.7	1,587.8	5.8	2.4	1,563.2	4.4	6.3	1,585.2	7.1
SH2	163.8	1,761.7	6.6	54.0	1,617.8	5.1	10.3	1,640.4	6.2	2.1	1,614.6	4.4	5.9	1,648.4	6.9
SH3	161.6	1,716.2	6.0	55.1	1,598.5	4.6	10.4	1,598.6	5.8	2.1	1,582.7	4.5	5.5	1,605.0	6.0
Moyenne	146.4	1,692.6	4.2	49.6	1,560.7	3.4	11.3	1,553.2	3.5	2.0	1,528.2	2.6	4.6	1,535.4	3.6

Conclusion

Ce mémoire aborde le problème de transport de patients dans un grand hôpital en Italie. Ce problème a été modélisé comme étant un problème de cueillettes et de livraisons dynamique. Pour résoudre ce problème, nous avons proposé trois approches de résolution différentes. La première approche statique utilise l'heuristique de l'ALNS comme méthode de résolution. Cette approche permet d'avoir les meilleurs résultats possibles, mais elle reste utopique et ne reflète pas la situation réelle de l'hôpital.

Pour cette raison, une deuxième approche dynamique, qui utilise une heuristique de réoptimisation, a été développée. Cet algorithme dynamique a été capable de construire des itinéraires pour les brancardiers en temps réel à chaque fois qu'une nouvelle demande de transport arrive. En plus de cela, nous avons développé une stratégie d'attente et deux stratégies de relocalisation afin d'optimiser les flux de transport. Ces stratégies sont basées sur des règles empiriques.

Avec la troisième heuristique, qui est une approche de planification basée sur des scénarios échantillonnés, nous avons utilisé l'historique des demandes de toute une année pour générer des scénarios qui peuvent se produire dans le futur proche et les résoudre. En se basant sur les résultats de ces scénarios, une fonction de consensus relocalise chaque véhicule vers l'endroit qui a la probabilité la plus élevée d'être la prochaine source de demande. Les résultats trouvés par cette approche sont meilleurs que ceux trouvés par l'heuristique de réoptimisation qui ne prend pas en considération des données stochastiques.

Le travail présenté dans ce mémoire peut-être encore amélioré en proposant un algorithme intelligent qui peut faire de meilleures prévisions de la demande. Même si nous avons basé nos prévisions sur toutes les données de l'année 2016, la tendance des demandes peut changer d'une année à une autre de façon remarquable. À titre d'exemple, nous pouvons citer la crise sanitaire de la Covid-19. De plus, notre méthode et les méthodes dans les travaux similaires supposent que la demande suit une loi normale, mais réellement, ce n'est pas toujours le cas. Parfois, la tendance de la demande peut changer à cause des facteurs extérieurs tels que la météo ou les grèves. L'approche statique et empirique confronte plusieurs défis tels que l'énorme nombre de données à traiter ainsi que le changement et l'évolution rapide des informations. Ces problèmes peuvent être résolus avec les algorithmes intelligents puisque le

traitement des données devient automatique et prend toujours les derniers événements en considération. Ces algorithmes nous permettent de prendre en considération des variables externes qui affectent la demande tels que les vacances et les autres événements connus. En utilisant l'intelligence artificielle, nous pouvons garantir que les prévisions suivent et s'ajustent aux dernières évolutions.

Bibliographie

- Maria BATTARRA, Jean-François CORDEAU et Manuel IORI : Chapter 6 : pickup-and-delivery problems for goods transportation. *In Vehicle Routing : Problems, Methods, and Applications, Second Edition*, pages 161–191. SIAM, 2014.
- Alexandre BEAUDRY, Gilbert LAPORTE, Teresa MELO et Stefan NICKEL : Dynamic transportation of patients in hospitals. *OR spectrum*, 32(1):77–107, 2010.
- Tolga BEKTAŞ, Panagiotis P REPOUSSIS et Christos D TARANTILIS : Chapter 11 : Dynamic vehicle routing problems. *In Vehicle Routing : Problems, Methods, and Applications, Second Edition*, pages 299–347. SIAM, 2014.
- Russell BENT et Pascal VAN HENTENRYCK : A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4):875–893, 2006.
- Russell BENT et Pascal VAN HENTENRYCK : Waiting and relocation strategies in online stochastic vehicle routing. *In IJCAI*, volume 7, pages 1816–1821, 2007.
- Russell W BENT et Pascal VAN HENTENRYCK : Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.
- Gerardo BERBEGLIA, Jean-François CORDEAU et Gilbert LAPORTE : Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15, 2010.
- Jean-François CÔTÉ, Thiago Alves de QUEIROZ, Francesco GALLESINI et Manuel IORI : Dynamic optimization algorithms for same-day delivery problems. 2021.
- George DANTZIG, Ray FULKERSON et Selmer JOHNSON : Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.
- George B DANTZIG et John H RAMSER : The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

- Paolo DETTI, Francesco PAPALINI et Garazi Zabalo Manrique de LARA : A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega*, 70:1–14, 2017.
- Karl F DOERNER et Juan-José SALAZAR-GONZÁLEZ : Chapter 7 : Pickup-and-delivery problems for people transportation. In *Vehicle Routing : Problems, Methods, and Applications, Second Edition*, pages 193–212. SIAM, 2014.
- Olive Jean DUNN : Multiple comparisons among means. *Journal of the American statistical association*, 56(293):52–64, 1961.
- Christian FIEGL et Carsten PONTOW : Online scheduling of pick-up and delivery tasks in hospitals. *Journal of Biomedical Informatics*, 42(4):624–632, 2009.
- Michel GENDREAU, Francois GUERTIN, Jean-Yves POTVIN et Éric TAILLARD : Parallel tabu search for real-time vehicle routing and dispatching. *Transportation science*, 33(4):381–390, 1999.
- Michel GENDREAU, Gilbert LAPORTE et René SÉGUIN : Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.
- Thomas HANNE, Teresa MELO et Stefan NICKEL : Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces*, 39(3):241–255, 2009.
- Sin C HO, Wai Yuen SZETO, Yong-Hong KUO, Janny MY LEUNG, Matthew PETERING et Terence WH TOU : A survey of dial-a-ride problems : Literature review and recent developments. *Transportation Research Part B : Methodological*, 111:395–421, 2018.
- Lars M HVATTUM, Arne LØKKETANGEN et Gilbert LAPORTE : Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, 40(4):421–438, 2006.
- Lars Magnus HVATTUM, Arne LØKKETANGEN et Gilbert LAPORTE : A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks : An International Journal*, 49(4):330–340, 2007.
- Bahman KALANTARI, Arthur V HILL et Sant R ARORA : An algorithm for the traveling salesman problem with pickup and delivery customers. *European Journal of Operational Research*, 22(3):377–386, 1985.
- Yannick KERGOSIEN, Ch LENTE, D PITON et J-C BILLAUT : A tabu search heuristic for the dynamic transportation of patients between care units. *European Journal of Operational Research*, 214(2):442–452, 2011.

- Alain LAFOREST : Des coûts en santé insoutenables, dit le ministre dubé, 2021. URL <https://www.journaldequebec.com/2021/08/22/des-couts-en-sante-insoutenables-dit-le-ministre-dube>.
- Sylvain LANDRY et Richard PHILIPPE : How logistics can service healthcare. *In Supply Chain Forum : An International Journal*, volume 5, pages 24–30. Taylor & Francis, 2004.
- John DC LITTLE, Katta G MURTY, Dura W SWEENEY et Caroline KAREL : An algorithm for the traveling salesman problem. *Operations research*, 11(6):972–989, 1963.
- Zhixing LUO, Mengyang LIU et Andrew LIM : A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. *Transportation Science*, 53(1):113–130, 2019.
- Oli BG MADSEN, Hans F RAVN et Jens Moberg RYGAARD : A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60(1):193–208, 1995.
- Emanuel MELACHRINOUDIS, Ahmet B ILHAN et Hokey MIN : A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 34(3):742–759, 2007.
- Snežana MITROVIĆ-MINIĆ et Gilbert LAPORTE : Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B : Methodological*, 38(7):635–655, 2004.
- Karen MOONS, Geert WAEYENBERGH et Liliane PINTELON : Measuring the logistics performance of internal hospital supply chains—a literature study. *Omega*, 82:205–217, 2019.
- Fredrik ODEGAARD, Li CHEN, Ryan QUEE et Martin L PUTERMAN : Improving the efficiency of hospital porter services, part 1 : study objectives and results. *Journal for Healthcare Quality*, 29(1):4–11, 2007a.
- Fredrik ODEGAARD, Li CHEN, Ryan QUEE et Martin L PUTERMAN : Improving the efficiency of hospital porter services, part 2 : schedule optimization and simulation model. *Journal for Healthcare Quality*, 29(1):12–18, 2007b.
- Victor PILLAC, Michel GENDREAU, Christelle GUÉRET et Andrés L MEDAGLIA : A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.
- Jean-Yves POTVIN et Jean-Marc ROUSSEAU : An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.
- Warren B POWELL : *Approximate Dynamic Programming : Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

- Ulrike RITZINGER, Jakob PUCHINGER et Richard F HARTL : A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1): 215–231, 2016.
- Stefan ROPKE et Jean-François CORDEAU : Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286, 2009.
- Stefan ROPKE et David PISINGER : An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006.
- Verena SCHMID et Karl F DOERNER : Examination and operating room scheduling including optimization of intrahospital routing. *Transportation Science*, 48(1):59–77, 2014.
- Michael SCHYNS : An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research*, 245(3):704–718, 2015.
- Paul SHAW : A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 46, 1997.
- Yongjia SONG, Marlin W ULMER, Barrett W THOMAS et Stein W WALLACE : Building trust in home services stochastic team-orienting with consistency constraints. *Transportation Science*, 54(3):823–838, 2020.
- P TOTH et D VIGO : The family of vehicle routing problem. *Vehicle Routing : Problems, Methods, and Applications*, pages 1–23, 2014.
- Marlin W ULMER, Justin C GOODSON, Dirk C MATTFELD et Barrett W THOMAS : On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics*, 9(2):100008, 2020.
- UNPD : La prochaine frontière : le développement humain et l’anthropocène, 2020. URL http://hdr.undp.org/sites/all/themes/hdr_theme/country-notes/fr/CAN.pdf.
- LJJ Van der BRUGGEN, Jan Karel LENSTRA et PC SCHUUR : Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science*, 27(3):298–311, 1993.
- Stacy A VOCCIA, Ann Melissa CAMPBELL et Barrett W THOMAS : The same-day delivery problem for online purchases. *Transportation Science*, 53(1):167–184, 2019.
- Stefan VONOLFEN et Michael AFFENZELLER : Distribution of waiting time for dynamic pickup and delivery problems. *Annals of Operations Research*, 236(2):359–382, 2016.

Frank WILCOXON : Individual comparisons by ranking methods. *In Breakthroughs in statistics*, pages 196–202. Springer, 1992.

The world BANK : Current health expenditure, 2018. URL data.worldbank.org.

Zhenzhen ZHANG, Mengyang LIU et Andrew LIM : A memetic algorithm for the patient transportation problem. *Omega*, 54:60–71, 2015.