



HAL
open science

Problèmes de tournées de véhicules pour la livraison de béton frais

Mohamed Seddik Hadjadj

► **To cite this version:**

Mohamed Seddik Hadjadj. Problèmes de tournées de véhicules pour la livraison de béton frais. Ingénierie assistée par ordinateur. Université de Lyon, 2021. Français. NNT : 2021LYSE1086 . tel-03901881

HAL Id: tel-03901881

<https://theses.hal.science/tel-03901881v1>

Submitted on 15 Dec 2022

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

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



Université Claude Bernard  Lyon 1

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de :

l'Université Claude Bernard Lyon 1

Ecole Doctorale ED 512

Informatique et Mathématiques (InfoMaths)

Spécialité de doctorat : Informatique

Soutenue publiquement le 02/06/2021, par :

Mohamed Seddik HADJADJ

Problèmes de tournées de véhicules pour la livraison de béton frais

Devant le jury composé de :

Mme. Salima HASSAS

Professeure, Université Lyon 1

M. Jean-Luc BARIL

Professeur, Université de Bourgogne

M. El-Ghazali TALBI

Professeur, Université de Lille

Mme. Zahia GUESSOUM

Maître de conférences HDR, Université de Reims Champagne-Ardenne

M. Hamamache KHEDDOUCI

Professeur, Université Lyon 1

M. Julien LEBOURGEOIS

CEO, Béton Direct

Présidente

Rapporteur

Rapporteur

Examinatrice

Directeur de thèse

Invité

Remerciements

Parce que je ne pouvais pas entamer l'exposé de mon travail sans rendre hommage à tous ceux qui m'ont aidé à l'accomplir, je profite de cette page pour les remercier et leur exprimer toute ma gratitude.

Mon directeur de thèse, le professeur HAMAMACHE KHEDDOUCI, est la première personne que j'aimerais remercier. Merci pour votre aide, pour les conseils que vous m'avez prodigués, et pour le temps précieux que vous m'avez consacré. Travailler avec vous fut pour moi une expérience très enrichissante.

À M. JULIEN LEBOURGEOIS, CEO de BÉTON DIRECT, entreprise au sein de laquelle j'ai effectué ma thèse, je tiens également à exprimer toute ma gratitude. La confiance que vous m'avez accordée, et votre disponibilité durant ces années de thèse, m'ont permis de travailler dans les meilleures conditions possibles.

Merci à toute l'équipe BÉTON DIRECT pour votre accueil chaleureux, votre bienveillance, et pour les moments agréables que j'ai passés parmi vous.

Merci aux membres de mon jury de soutenance d'avoir pris le temps d'examiner mon travail. J'espère que vous avez pris autant de plaisir à lire ce document que moi à l'écrire.

Merci à ma famille et à mes amis pour votre présence à mes côtés et votre soutien indéfectible.

Merci, enfin, à tous ceux qui, de près ou de loin, ont contribué à l'accomplissement de ce travail, je vous en suis infiniment reconnaissant.

Résumé

Nous nous intéressons dans ce travail à la construction de tournées de véhicules pour une entreprise spécialisée dans la vente et la livraison de béton frais.

Deux schémas logistiques sont étudiés. Dans le premier, un unique véhicule doit livrer des bacs remplis de béton à des clients, tout en collectant des bacs vides livrés lors de précédentes tournées. De plus, le véhicule peut être amené à effectuer des détours imprévus en cours de route. Il s'agit donc de tournées de véhicules avec collectes et livraisons, dans un contexte dynamique et incertain.

Pour sa part, le second schéma logistique fait appel à deux véhicules. Le premier livre et collecte des bacs vides, tandis que le second passe dans la foulée du premier pour remplir les bacs fraîchement livrés. Ce schéma décrit des tournées de véhicules avec contraintes de synchronisation et de précedence entre les deux véhicules considérés.

Pour le premier schéma logistique, nous développons et comparons deux approches dont le but est de construire des tournées efficaces capables de s'adapter aux détours imprévus qui peuvent se produire en cours de route.

Pour le second schéma logistique, deux approches sont également proposées et testées, l'objectif étant de générer des tournées performantes qui synchronisent les visites des deux véhicules chez les clients.

Mots-clés : Tournées de véhicules avec livraisons et collectes, Tournées de véhicules stochastiques/dynamiques, Tournées synchronisées, Métaheuristiques...

Abstract

In this work, we consider vehicle routing problems for the delivery of ready-mixed concrete.

Two different types of VRP's have been studied. The first type involves a single vehicle which has to visit n customers in order to deliver small containers of ready-mixed concrete or pick them up after they have been used. Unexpected detours to recycling centres are considered, which makes tours uncertain beforehand and involves dealing with stochastic and dynamic tours. Such vehicle routing problems are known as stochastic and/or dynamic VRP's with pickup and delivery.

On the other hand, the second type of VRP's studied in this work involves two different vehicles. The first has to visit n customers in order to pick up or deliver empty containers, while the second has to deliver ready-mixed concrete by pouring it into the previously delivered containers. This problem is a VRP with synchronization and temporal precedence constraints between two vehicles.

For the first problem, we propose two approaches in order to build efficient vehicle routes that can adapt to unplanned detours that may occur during the tours.

Two approaches have also been studied for the second problem in order to generate effective synchronized routes for the two vehicles considered in the tours.

Keywords : Vehicle routing problems with Pickup & Delivery, Dynamic/Stochastic vehicle routing problems, Synchronization constraints, Metaheuristic...

Table des matières

Introduction générale	1
1 Contexte de la thèse	3
Introduction	3
1.1 Présentation de l'entreprise	3
1.2 Commander du béton prêt à l'emploi	4
1.2.1 Identifier la quantité et le type de béton nécessaires	4
1.2.2 Identifier la solution logistique appropriée	5
1.3 Le parcours d'achat "classique" de BPE	9
1.4 Le parcours d'achat "Béton Direct"	11
1.5 Problématique	13
Conclusion	19
2 Schéma logistique 1 : Tournées de livraison et de collecte à véhicule unique	21
Introduction	21
2.1 Description du schéma logistique	22
2.1.1 Point de vue client	22
2.1.2 Point de vue Béton Direct	22
2.1.3 Recyclage du béton non consommé	23
2.2 Sujets de recherche sous-jacents	26
2.3 Tournées de véhicules avec collectes & livraisons	28
2.3.1 Classification	28
2.3.2 Revue de la littérature	29

2.3.3	Synthèse	40
2.4	Tournées de véhicules stochastiques/dynamiques	41
2.4.1	Classification	41
2.4.2	Sources de dynamisme	43
2.4.3	Approches de résolution	45
2.4.4	Synthèse	48
2.5	Formalisation du problème	50
2.6	Contribution	53
2.7	Approche stochastique	55
2.7.1	Principe	55
2.7.2	Définition de la solution	57
2.7.3	Choix des solutions initiales	57
2.7.4	Définition du voisinage d'une solution	61
2.7.5	Cohérence d'une solution	62
2.7.6	Fonction objectif	63
2.7.7	Solution "à priori" VS solution "à postériori"	65
2.7.8	Estimateur de la fonction objectif	66
2.7.9	Choix du centre de recyclage	67
2.7.10	Résumé	67
2.8	Approche dynamique	69
2.8.1	Approche stochastique VS Approche dynamique	70
2.8.2	Principe	70
2.9	Expérimentations	73
2.9.1	Implémentation	74
2.9.2	Jeu de données	74
2.9.3	Résultats	75
	Conclusion	79
3	Schéma logistique 2 : Tournées avec deux véhicules synchronisés	83
	Introduction	83
3.1	Description du schéma logistique	83
3.1.1	Point de vue client	84
3.1.2	Point de vue de l'entreprise	84

3.2	Schéma 1 VS Schéma 2	87
3.3	Tournées de véhicules synchronisées	88
3.4	Formalisation du problème	90
3.5	Contribution	95
3.6	Optimisation séquentielle	96
3.6.1	Construction de la tournée de livraison et de collecte de bacs	96
3.6.2	Construction de la tournée de livraison de béton	96
3.7	Optimisation simultanée	98
3.7.1	Principe	98
3.7.2	Représentation d'une solution	99
3.7.3	Définition du voisinage	100
3.8	Expérimentations	103
3.8.1	Implémentation	103
3.8.2	Jeu de données	103
3.8.3	Résultats	104
	Conclusion	108
	Conclusion générale et perspectives	111
	Liste des publications	115
	Bibliographie	117

Table des figures

1.1	Camion toupie	5
1.2	Chantier sans obstacle VS avec obstacle	6
1.3	Camion toupie équipé d'une pompe (pumi)	7
1.4	Schéma illustrant l'utilisation d'une toupie équipée d'une pompe	7
1.5	Camion toupie équipé d'un tapis	8
1.6	Schéma illustrant l'utilisation d'une toupie équipée d'un tapis	8
1.7	Parcours d'achat traditionnel de béton prêt à l'emploi	10
1.8	Parcours d'achat BÉTON DIRECT	12
1.9	4 Bacs à béton pleins	16
1.10	Camion grue	18
1.11	Véhicule utilitaire	18
2.1	Le schéma logistique n° 1 vu par le client	23
2.2	Le schéma logistique n° 1 vu par BÉTON DIRECT	24
2.3	Le schéma logistique n° 1 vu par le véhicule effectuant la tournée	26
2.4	Tournée de livraison et de collecte de bacs avec détours en centres de recyclage	27
2.5	Classification des <i>Pickup & Delivery Problems</i> (PDP) selon [Berbeglia et al., 2007]	40
2.6	Classification des VRP selon la qualité et l'évolution des données [Pillac et al., 2013]	49
2.7	Graphe représentant une tournée de livraison et de collecte de bacs avec détours en centres de recyclage	54
2.8	Représentation d'une tournée	57

2.9	Solution incohérente VS Solution cohérente	60
2.10	Heuristique 3 : Alternance livraison/collecte	61
2.11	Algorithme <i>1-Shift</i> : principe	62
2.12	Algorithme <i>1-Shift</i> : contrainte de capacité	64
2.13	Solution "à priori" VS solution "à postérieure"	66
2.14	Choix du centre de recyclage	68
2.15	Approche stochastique : vue globale	69
2.16	Approche dynamique	73
2.17	Part des détours dans le coût moyen global des solutions trouvées	78
3.1	Le schéma logistique n° 2 vu par le client	85
3.2	Le schéma logistique n° 2 vu par l'entreprise	86
3.3	Construction d'une tournée de livraison de béton	97
3.4	Représentation des tournées	101
3.5	Génération d'une solution voisine	102
3.6	Résumé des schémas et approches étudiés	113

Liste des tableaux

- 2.1 Approche stochastique VS Approche dynamique 70
- 2.2 Coût global moyen des solutions obtenues avec l’approche stochastique sur les instances de [Gendreau et al., 1999] 76
- 2.3 Taux d’augmentation du coût moyen des solutions obtenues lorsque la probabilité de survenue de détours $p = 0,9$, par rapport au cas où $p = 0,1$ 78
- 2.4 Coût global moyen des solutions obtenues avec l’approche stochastique et l’approche dynamique sur les instances de [Gendreau et al., 1999] 80

- 3.1 Coût moyen (ou durées de trajet moyennes) des solutions obtenues avec la stratégie d’optimisation séquentielle sur les instances de [Gendreau et al., 1999] 104
- 3.2 Coût moyen (ou durées de trajet moyennes) des solutions obtenues avec la stratégie d’optimisation simultanée sur les instances de [Gendreau et al., 1999] 105
- 3.3 Optimisation simultanée VS Optimisation séquentielle 107
- 3.4 Solutions optimales VS Optimisation simultanée 107

Introduction générale

Nombreux sont les secteurs d'activités où l'on retrouve, d'une manière ou d'une autre, le besoin d'organiser des tournées de véhicules. Industrie, santé, transport de biens ou de personnes en sont des exemples parmi tant d'autres. Aussi, ce besoin omniprésent a poussé les chercheurs à s'intéresser à ce type de problèmes, et cela se traduit depuis des décennies par un nombre incalculable de travaux de recherche, traitant de problèmes divers et variés. Différentes contraintes et différents objectifs ont été étudiés, si bien que les problèmes de tournées de véhicules (VRP) forment aujourd'hui une des classes de problèmes de recherche opérationnelle les plus étudiées, et les plus célèbres.

Avec la démocratisation du e-commerce, et tout ce que cela implique comme nouveaux enjeux en termes de logistique de livraison, l'importance des VRP n'est pas près de diminuer, d'autant plus qu'un nombre croissant d'entreprises ces dernières années voient dans l'optimisation des tournées de véhicules un moyen de réduire leur empreinte carbone, et il est très probable que cette tendance s'accroisse encore plus dans les années qui viennent au vu de la place, de plus en plus importante, que sont en train de prendre les sujets liés à l'écologie et à l'environnement. Le travail effectué dans le cadre de cette thèse illustre un exemple concret d'une entreprise qui fait du e-commerce, et qui souhaite optimiser ses tournées de véhicules en vue de réduire ses coûts (pour proposer de meilleurs prix à ses clients), et de diminuer l'impact environnemental dû à son activité.

BÉTON DIRECT, l'entreprise avec laquelle ce travail est fait en collaboration, est une startup spécialisée dans la vente en ligne et la livraison de béton prêt à l'emploi. S'adressant principalement à une clientèle de particuliers, petits consommateurs de béton, l'en-

treprise a constaté que les modes de livraison existants pour ce matériau n'étaient pas adaptés aux petites quantités. En effet, quelle que soit la quantité de béton livrée, ce sont toujours les mêmes types de véhicules qui sont utilisés par les livreurs, de gros camions lourds, encombrants, polluants, et bien trop coûteux lorsqu'il s'agit de livrer des petites quantités.

L'idée de BÉTON DIRECT pour répondre à ce problème est d'exploiter la notion de tournées de véhicules. Ainsi, en mutualisant les matériels de transport pour plusieurs clients demandant de petits volumes de béton au lieu de mobiliser un véhicule lourd pour chaque commande, les coûts de transport par client sont réduits, de même que l'empreinte CO2 correspondante. Pour ce faire, l'entreprise a besoin d'un outil d'aide à la décision pour optimiser ces tournées de véhicules, d'où la collaboration avec un laboratoire de recherche qui a donné naissance à ce travail de thèse.

La suite de ce document est structurée comme suit :

- Le premier chapitre décrit le contexte de la thèse. Entre autres, on y présente l'entreprise BÉTON DIRECT, ses missions, ainsi que les principales problématiques auxquelles elle se trouve confrontée. Certaines notions liées à la vente et à livraison de béton sont également présentées afin de mieux comprendre le contexte de ce travail.
- Le deuxième chapitre se penche sur un premier type de tournées envisagé pour la livraison de petites quantités de béton. Ce premier schéma logistique est d'abord décrit, puis formalisé. Une revue de la littérature impliquant des tournées similaires est également exposée, avant que des approches de résolution ne soient proposées et testées à la fin du chapitre.
- De la même manière que pour le chapitre 2, le troisième s'intéresse à un second type de tournées envisagé pour la livraison de béton en petites quantités. Une description détaillée, une revue de la littérature associée, ainsi qu'une formalisation du problème sont présentées. Des approches de résolution sont proposées et comparées en fin de chapitre.

Contexte de la thèse

Introduction

Afin de comprendre la problématique sur laquelle nous nous sommes penchés, il est nécessaire de connaître le contexte dans lequel elle est née et les enjeux qui l'entourent.

Ce chapitre a pour objectif de décrire le contexte de la présente thèse. Il inclut notamment :

- une présentation de BÉTON DIRECT, l'entreprise avec laquelle ce travail a été effectué. Son activité, ses missions et ses objectifs y sont passés en revue ;
- un résumé de certains concepts de base, notamment liés à la vente et à la livraison de béton prêt à l'emploi ;
- une description de la problématique soulevée par l'entreprise, et qui a donné naissance à ce travail.

1.1 Présentation de l'entreprise

BÉTON DIRECT est une entreprise Lyonnaise créée en 2015, spécialisée dans la vente en ligne de béton prêt à l'emploi (BPE), et s'adressant principalement à une clientèle de petits consommateurs et consommateurs occasionnels (en d'autres termes, les particuliers et les petits artisans).

Dans un secteur conservateur et inerte, où les pratiques n'ont pas beaucoup changé ces dernières années, et où les nouvelles technologies ne sont que très peu exploitées, BÉTON DIRECT s'est lancé dans le développement d'outils innovants, visant à simplifier le parcours d'achat de BPE pour sa clientèle cible, à lui proposer les meilleurs tarifs possibles,

et à intégrer une part d'automatisation dans le métier de commercial du béton.

1.2 Commander du béton prêt à l'emploi

Afin de mesurer la valeur ajoutée de BÉTON DIRECT pour un client voulant commander du BPE, il est important de comprendre ce qu'implique concrètement une livraison de béton.

1.2.1 Identifier la quantité et le type de béton nécessaires

Pour commander du BPE, il faut d'abord identifier le type et la quantité de béton adéquats pour la réalisation de l'ouvrage voulu. Pour ce qui est de la quantité, il s'agit de déterminer le **volume** nécessaire et suffisant (en m^3) pour réaliser l'ouvrage, en respectant les règles de l'art, et en évitant les gaspillages.

Ensuite, s'agissant du type de béton, il faut savoir qu'il existe des dizaines de "recettes" différentes pour faire du béton (on parle de **formules**). Chaque formule est caractérisée par un certain nombre de propriétés (classe de consistance, classe de résistance, classe d'exposition...). La formule idoine est choisie en fonction de trois principaux paramètres :

- le type d'ouvrage désiré. Parmi les ouvrages les plus fréquents chez les clients particuliers, on trouve, entre autres : la dalle de terrasse, la piscine, le mur de clôture, la descente de garage...

Certains ouvrages auront besoin d'un béton plus résistant (pour supporter des poids importants par exemple), d'autres nécessiteront un béton résistant à l'eau (pour une piscine notamment)...

- les conditions climatiques le jour du **coulage** du béton. Il existe notamment des formules pouvant retarder son durcissement de quelques heures, de quoi permettre au client de l'utiliser plus sereinement lorsque la température est élevée, et que le risque de durcissement rapide est plus grand...
- le niveau de confort que le client souhaite pour la mise en œuvre de son béton. Certaines formules permettent en effet de fluidifier le matériau pour le rendre plus facile à manipuler et à placer (le béton étant, de base, un matériau très lourd et difficile à manier).

1.2.2 Identifier la solution logistique appropriée

Une fois la formule de béton choisie et le volume fixé, l'étape suivante consiste à définir la solution logistique convenable pour livrer le béton sur chantier, dans les meilleures conditions possibles.

La solution logistique comprend :

a) Le matériel alloué au transport du béton

En d'autres termes, le(s) camion(s) nécessaire(s) pour transporter le béton de la centrale où il a été fabriqué, vers le chantier du client. Le BPE se transporte, en général, dans des camions spéciaux dotés de réservoirs rotatifs permettant le brassage continu du béton pendant le transport. On parle de **camion malaxeur**, de **camion toupie**, ou simplement de **toupie** (voir figure 1.1). Le camion toupie a une capacité maximale définie. Le nombre de camions envoyés chez le client dépend donc du volume à transporter et de la capacité des camions à disposition (le volume maximal transporté par une toupie varie entre 4 et $10m^3$).



FIGURE 1.1 – Camion toupie

b) Le matériel de manutention

Dans certains cas (une livraison sur 2 environ pour les clients BÉTON DIRECT), la configuration du chantier du client ne permet pas de couler le béton directement à partir du camion toupie qui le transporte. Cela arrive généralement lorsqu'un **obstacle** se dresse devant le point de coulage prévu (c'est-à-dire, devant l'emplacement de la future dalle, piscine...), empêchant le camion toupie de se rapprocher suffisamment pour déverser le

béton là où il faut, et créant ainsi une **distance** plus ou moins importante entre le sus-mentionné **point de coulage et le point de stationnement du camion** (voir figure 1.2). L'obstacle dont il est question ici peut être de différentes natures. Il s'agit de tout ce qui peut empêcher un camion d'avancer vers le point de coulage :

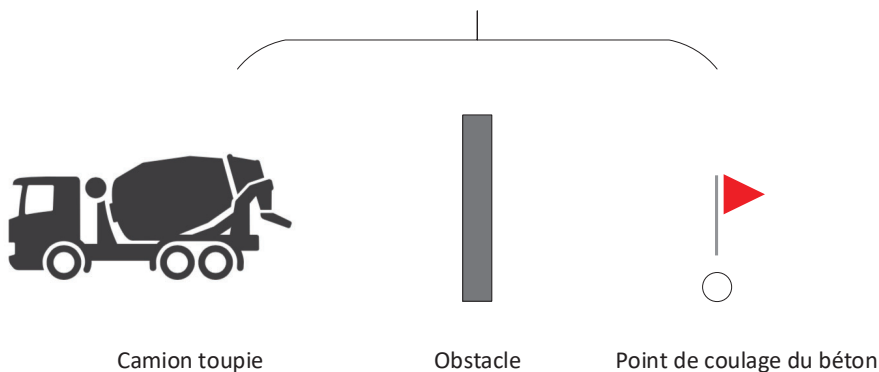
- un mur ;
- une haie ;
- un arbre ;
- une ruelle trop étroite pour le camion ;
- des fils électriques qui empêchent le véhicule de passer...



Le béton est déversé directement du camion vers le point de coulage

(a) Chantier sans obstacle

Distance entre le point de coulage et le point de stationnement du camion



(b) Chantier avec obstacle

FIGURE 1.2 – Chantier sans obstacle VS avec obstacle

Dans de tels cas, la solution logistique recommandée pour livrer le béton doit intégrer, en plus des camions toupies nécessaires au transport, un matériel de manutention spécifique, qui, une fois les toupies chez le client, servira à amener le béton jusqu'au point de coulage. Il existe 2 principaux types de matériels de manutention :

la pompe : montée sur un véhicule spécifique, ou directement intégrée au camion

toupie (on parle alors d'un **camion malaxeur pompe**, ou **pumi**), elle permet de pomper le béton, à partir de la toupie, à travers un long bras articulé capable de contourner les obstacles, pour le déverser à l'endroit désiré (voir figure 1.4). Les pompes à béton diffèrent selon la longueur du bras articulé qui leur est associé (la longueur du bras varie entre 16 et 42m et le choix de la pompe adéquate pour un chantier se fait d'après la distance entre le point de coulage et le point de stationnement du camion ;



FIGURE 1.3 – Camion toupie équipé d'une pompe (pumi)

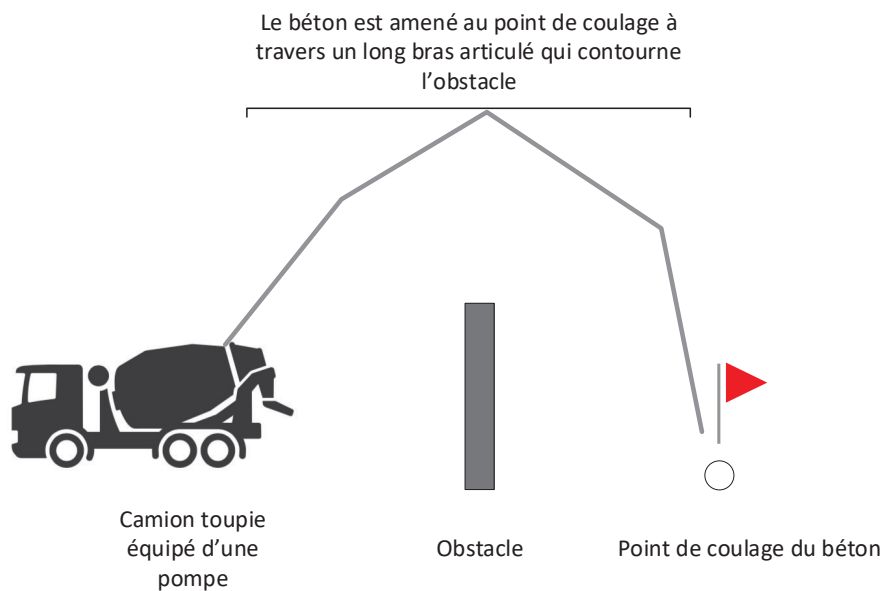


FIGURE 1.4 – Schéma illustrant l'utilisation d'une toupie équipée d'une pompe

le tapis : il s'agit d'une bande convoyeuse incorporée dans certains camions toupies (on parle alors de **camion toupie avec tapis**), qui permet de transporter le béton à sa sortie de la toupie vers le point de coulage, à la manière d'un tapis roulant à

bagages dans un aéroport (voir figure 1.6). Là encore, il existe différentes longueurs possibles pour un tapis, le choix de la solution adéquate se faisant en fonction de la configuration du chantier.



FIGURE 1.5 – Camion toupie équipé d'un tapis

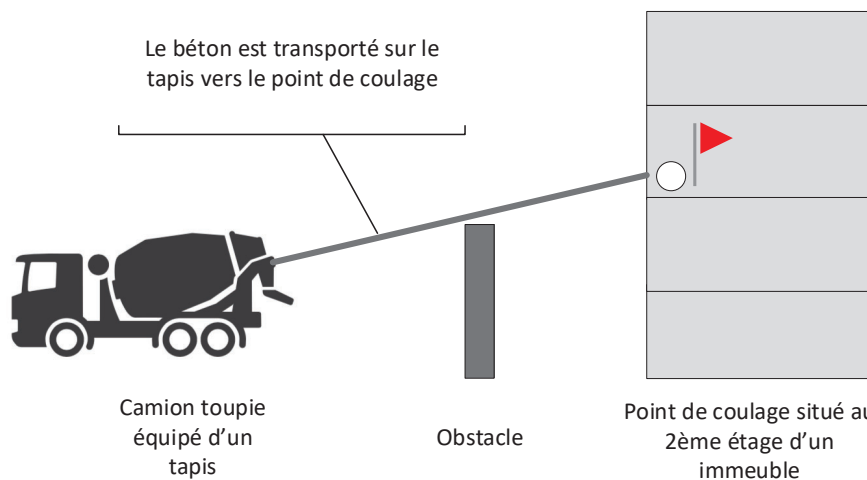


FIGURE 1.6 – Schéma illustrant l'utilisation d'une toupie équipée d'un tapis

Pour résumer, avant de commander du béton prêt à l'emploi, il faut choisir le type et la quantité de béton nécessaires ainsi que la solution logistique adaptée. Lorsqu'il achète du BPE, un client paye :

1. le béton ;
2. la prestation de transport ;
3. et, éventuellement, la prestation de manutention (pompage, tapissage...).

1.3 Le parcours d'achat "classique" de BPE

La clientèle de particuliers (la cible privilégiée de BÉTON DIRECT) se caractérise très souvent par :

- de faibles volumes commandés** : le particulier a souvent besoin de béton pour de petits travaux chez lui (une dalle pour sa terrasse, une piscine, un abri de jardin...). Ce genre d'ouvrages ne nécessite pas de gros volumes de béton (le volume moyen commandé par un client particulier chez BÉTON DIRECT se situe entre 5 et $6m^3$);
- un faible taux de récurrence** : le besoin en béton du particulier est très souvent ponctuel, il n'a pas vocation à commander régulièrement.

Aussi, cette clientèle est souvent délaissée par les producteurs de béton, qui préfèrent concentrer leur activité sur des consommateurs plus réguliers, avec de plus gros besoins. Ce manque d'intérêt se manifeste notamment dans le parcours d'achat "traditionnel" de BPE, qui peut s'avérer long et fastidieux pour le petit consommateur qui veut commander du béton.

D'abord, le client doit identifier les centrales productrices de béton se trouvant à proximité de son chantier. En effet, le BPE étant un matériau frais, il durcit relativement vite après fabrication. Cela oblige les centrales à restreindre leurs zones de livraison pour faire parvenir le produit chez le client dans un état permettant une utilisation optimale.

Une fois les centrales à proximité du chantier identifiées, le client doit les contacter une à une, par téléphone généralement, pour essayer d'obtenir des devis, et les commerciaux n'ont pas toujours le temps de s'occuper de petites livraisons, surtout en période de haute saison.

De plus, pour confirmer la commande, le contact téléphonique entre le client et la centrale à béton ne suffit pas. Il est souvent nécessaire de faire déplacer un commercial pour vérifier, de visu, les conditions d'accès au chantier (présence d'obstacles, de fils électriques, largeur de la rue où doit passer le camion...) et définir ainsi la solution logistique adéquate (voir section 1.2). Pour se faire livrer du béton, le client doit donc faire le tour du marché pour trouver une centrale capable de le livrer, à un prix correct, et attendre le passage du commercial de cette centrale pour définir la logistique de livraison. Ce processus long décourage les consommateurs, les oblige à s'y prendre longtemps à l'avance pour être sûrs d'être livrés à temps, et les empêche parfois de faire le tour du marché pour trouver le

meilleur prix possible.

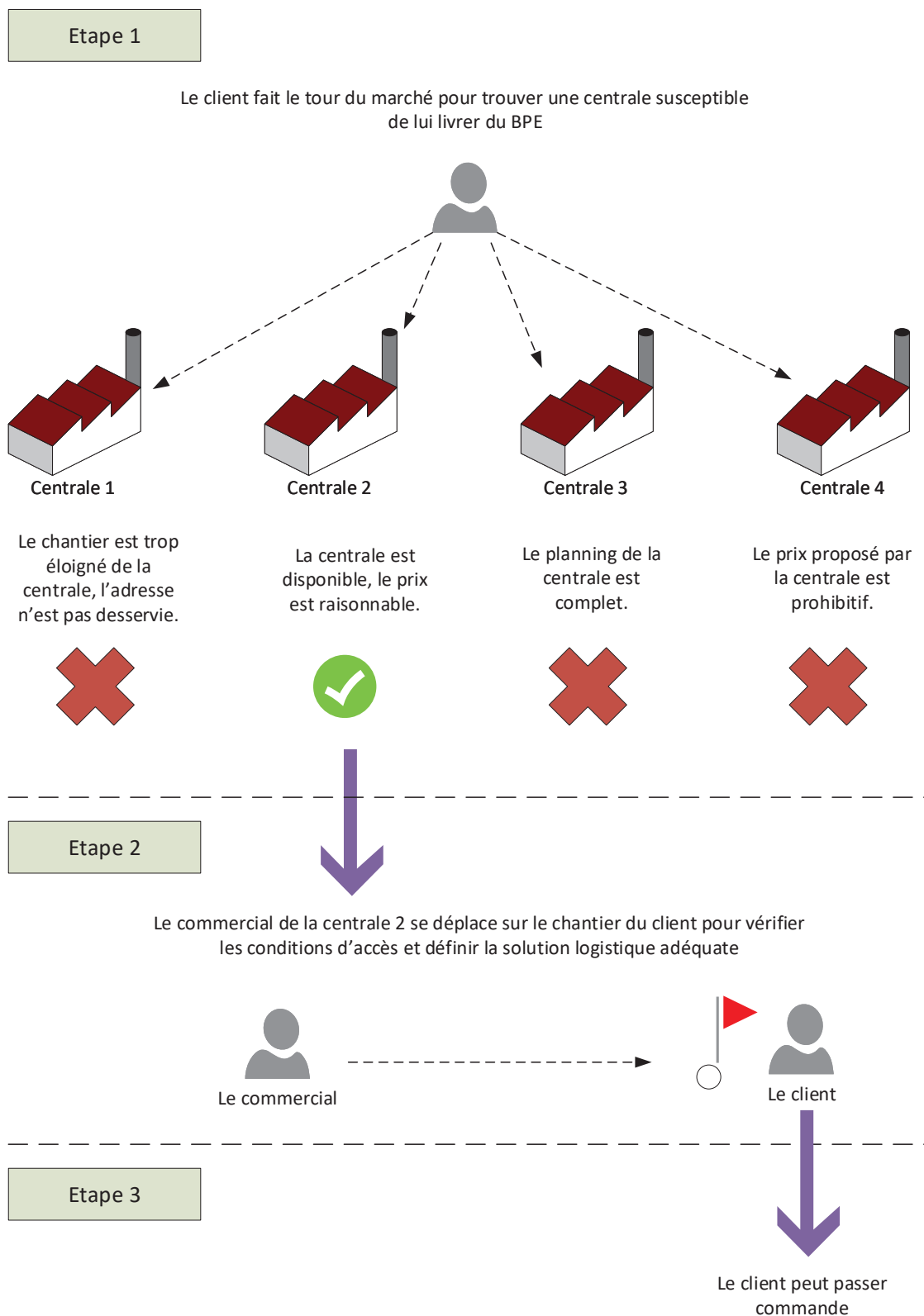


FIGURE 1.7 – Parcours d'achat traditionnel de béton prêt à l'emploi

1.4 Le parcours d'achat "Béton Direct"

Pour un client qui passe par BÉTON DIRECT, plus besoin de faire le tour du marché pour obtenir des devis chez les différents producteurs de béton. Le client n'a qu'un seul interlocuteur, BÉTON DIRECT, via son site web. Il y décrit son projet en remplissant un formulaire :

- adresse du chantier ;
- type d'ouvrage souhaité (dalle, piscine...);
- volume nécessaire ;
- conditions d'accès au chantier. Une carte *Google Maps* montrant une vue aérienne du chantier est présentée au client. Il doit y indiquer à l'aide de curseurs, les points de coulage et de stationnement du camion prévus, et la présence éventuelle d'obstacles ;

Une fois le formulaire soumis, et à partir des informations qu'il contient, un algorithme se charge de générer un devis automatique visible immédiatement par le client. Le travail de l'algorithme consiste alors à :

1. trouver les centrales à béton à proximité du chantier. BÉTON DIRECT travaille avec un réseau de producteurs dans toute la France, composé d'environ 250 centrales à béton ;
2. choisir la centrale à béton qui offre le prix le plus intéressant ;
3. choisir la bonne solution logistique pour la livraison ;
4. calculer un prix de vente comprenant le prix d'achat chez la centrale considérée + une marge commerciale (le bénéfice) pour BÉTON DIRECT.

Une fois le devis créé, un conseiller BÉTON DIRECT procède à sa "validation". Il s'agit de vérifier que les renseignements saisis par le client sont corrects, et apporter des modifications au devis le cas échéant. Cette étape nécessite souvent pour le conseiller BÉTON DIRECT de joindre le client par téléphone. Lorsque le devis est validé, le client peut passer commande sur le site. La suite est transparente pour le client, BÉTON DIRECT contacte la centrale choisie pour confirmer la livraison.

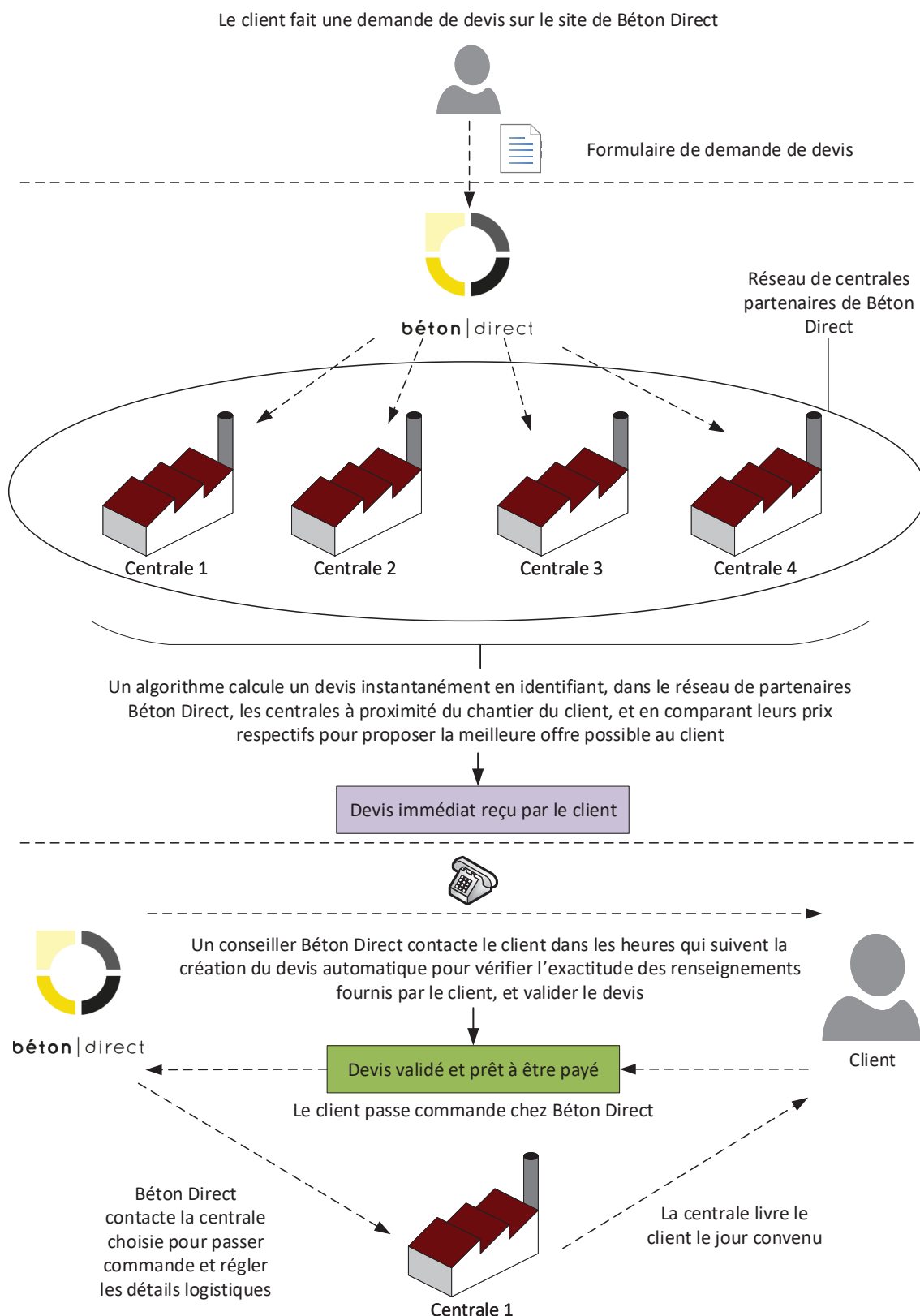


FIGURE 1.8 – Parcours d'achat BÉTON DIRECT

Du point de vue du client, le parcours d'achat BÉTON DIRECT permet de gagner

en simplicité et en rapidité. Le client obtient un devis immédiat, prenant en compte les prix de plusieurs producteurs concurrents, et cela en passant par un interlocuteur unique. Cet interlocuteur unique s'occupe de la logistique de livraison, en faisant l'économie de déplacements coûteux et chronophages sur chantier, le choix de la logistique se faisant à distance, en prenant en compte la vue aérienne du chantier du client (GOOGLE MAPS). En plus de simplifier les démarches du client, la plateforme BÉTON DIRECT permet également de faire évoluer le métier de commercial. En effet, là où le commercial "traditionnel" doit prendre les appels des clients et leur faire des devis "à la main", ce qui peut lui prendre beaucoup de temps, l'algorithme développé par BÉTON DIRECT permet de générer des centaines de devis par jour.

1.5 Problématique

En faisant des petits consommateurs sa cible privilégiée, BÉTON DIRECT a identifié deux principaux axes d'amélioration dans l'expérience d'achat de BPE pour sa clientèle cible.

D'abord, le parcours d'achat en lui-même. Comme cela a été expliqué dans les précédentes sections, le processus classique permettant au client de commander du béton prêt à l'emploi manque de fluidité et de simplicité. BÉTON DIRECT s'est attaqué à ce problème en s'appuyant sur les nouvelles technologies, bien trop peu exploitées par ailleurs dans le monde du BPE. Le développement d'outils numériques permet d'injecter une bonne dose d'automatisation, et donc de fluidifier et de simplifier un parcours d'achat autrefois long, fastidieux, et pas toujours très clair.

Parallèlement à ce premier sujet important que représente l'automatisation du parcours d'achat, un deuxième axe d'amélioration de l'expérience client a été identifié très rapidement par la startup.

En effet, du fait du profil de sa clientèle cible, l'entreprise est souvent amenée à traiter des commandes impliquant de très faibles volumes de béton, ne dépassant pas les $2m^3$. Cela a très vite permis de mettre en lumière un sujet rarement évoqué, souvent négligé, et jamais réellement traité : l'inadéquation des solutions logistiques proposées pour la livraison de BPE lorsque le volume livré est faible (on dira qu'un volume est "faible" lorsqu'il est inférieur ou égal à $2m^3$).

Par solution logistique, on entend :

- l'ensemble des camions toupies nécessaires pour transporter le béton de la centrale où il est fabriqué vers le chantier du client ;
- éventuellement, le matériel de manutention nécessaire (pompes, tapis...) pour transporter le béton du camion où il a été transporté, vers le point de coulage (voir section 1.2 pour plus de détails).

Dans la grande majorité des cas, ce sont les producteurs eux-mêmes qui fournissent la prestation de livraison du béton qu'ils fabriquent. Ils possèdent, pour la plupart, leurs propres flottes de camions toupies, qu'ils constituent en prenant en compte les besoins de leur clientèle cible, les gros consommateurs. Ainsi, les producteurs privilégient les camions toupies ayant une grande capacité (pouvant transporter entre 6 et $8m^3$) pour livrer leurs clients.

Étant donné que les producteurs souhaitent toujours maximiser l'utilisation de leurs camions afin de les rentabiliser, et que les petits consommateurs de béton ne font pas partie de leurs priorités, ils ne voient pas d'intérêt à investir dans le développement de solutions logistiques mieux adaptées à la livraison de petites quantités. Par conséquent, un client qui commandera $1m^3$ recevra très souvent son produit dans un camion qui pourrait servir à livrer 6 ou $7m^3$.

Cela pose plusieurs problèmes :

- pour le client d'abord, le coût de transport qui lui est facturé dans ces cas-là est disproportionné par rapport au coût du produit qu'il aura acheté. De plus, pour peu qu'il y ait besoin d'un matériel de manutention supplémentaire une fois le camion sur chantier, le client peut vite se retrouver avec une facture dont le prix de la logistique représente plus de la moitié du coût total, ce qui est absurde, et souvent rédhibitoire pour le client ;
- d'un point de vue écologique ensuite, il va sans dire que le modèle actuel qui consiste à mobiliser un camion toupie énergivore et émetteur de gaz à effet de serre à chaque commande, aussi petite soit-elle, a un impact environnemental non négligeable, et doit sans doute être remis en question.

Le constat ainsi dressé a poussé BÉTON DIRECT à explorer de nouvelles pistes en matière de solutions de livraison mieux adaptées aux petites quantités et plus avantageuses en termes de prix, et d'empreinte carbone. Le recours à la mutualisation des moyens de

transport a très vite été envisagé comme une façon de minimiser les coûts, ainsi que l'impact environnemental liés au transport du béton.

En effet, au lieu de mobiliser un camion pour chaque petite livraison de béton, pourquoi ne pas organiser des tournées de véhicules durant lesquelles plusieurs petits consommateurs seraient livrés le même jour, par le même véhicule. Cela permettrait d'envoyer une seule toupie pleine livrer n petits consommateurs, au lieu d'envoyer (comme c'est majoritairement le cas), n toupies quasi vides, chez n clients.

Planifier des tournées de toupies pour optimiser les différents coûts de transport est une idée de bon sens qui n'a jamais été mise en œuvre dans le passé, ou du moins, jamais de façon systématique et automatique. Néanmoins, le béton prêt à l'emploi étant un matériau frais, sa livraison obéit à certaines contraintes rendant la simple organisation de tournées de toupies insuffisante pour garantir un béton livré dans les meilleures conditions.

D'abord, envoyer une toupie faire le tour d'un ensemble de points de livraison pour déposer la quantité de béton commandée par chaque client est une tâche délicate dans le sens où les toupies circulant de nos jours ne disposent pas d'une "jauge" capable d'indiquer la quantité de béton restant à l'intérieur de la toupie. Il est donc impossible, en arrivant chez un client au cours d'une tournée, de déverser la quantité exacte de béton qu'il a demandée, car il n'y a aucun indicateur, au sein de la toupie, capable de mesurer le volume déversé, ou le volume restant dans le camion.

Par ailleurs, un autre problème pratique se pose dès lors que l'on souhaite construire des tournées de toupies pour livrer du béton. Dans le cas où le point de coulage (c'est-à-dire, le point où le béton est censé être déversé) n'est pas accessible par la toupie (par exemple, à cause de la présence d'un obstacle empêchant la toupie de se rapprocher suffisamment du point de coulage), le recours à du matériel de manutention peut s'avérer nécessaire (voir section 1.2.2). Il faudrait alors faire intervenir une pompe ou un tapis chez le client pour transporter le béton du camion jusqu'au point de coulage, ce qui n'est pas envisageable pour deux raisons principales :

1. le coût que cela engendrerait serait disproportionné par rapport à la faible quantité de béton commandée. L'objectif visé par BÉTON DIRECT est justement de réduire les coûts liés à la logistique pour les petites commandes, ce qui inclut le coût du transport d'une part, et le coût de la manutention d'autre part ;
2. le recours à du matériel de manutention implique l'immobilisation de la toupie

sur le chantier le temps de couler tout le béton. En effet, lorsqu'une toupie arrive sur un chantier et qu'elle ne peut pas accéder au point de coulage, le béton qu'elle transporte est pompé (dans le cas où l'on fait intervenir une pompe), ou transporté sur une bande transporteuse (dans le cas d'un recours à un tapis). Dans les deux cas, la toupie doit être immobilisée sur le chantier tout le temps que dure l'opération, ce qui représente une perte de temps considérable. Dans de telles conditions, la toupie ne pourra effectuer qu'un nombre limité de livraisons car elle sera très longtemps immobilisée sur les différents chantiers.

Pour mettre en œuvre un système de tournées de livraison de béton frais, il faut donc répondre aux deux questions soulevées ci-dessus, à savoir :

- au cours d'une tournée, comment livrer, pour chaque client, la quantité exacte qu'il a commandée, sachant qu'il n'y a aucun dispositif intégré dans la toupie qui permette de mesurer le volume de béton déversé, ou le volume restant ?
- au cours d'une tournée, comment traiter la problème de la manutention. En d'autres termes, comment faire en sorte de ne pas immobiliser la toupie sur le chantier pendant toute la durée du coulage ? Comment permettre à la toupie de déverser le béton rapidement à chaque point de livraison pour, ainsi, minimiser la durée d'attente du camion sur le chantier, et rentabiliser les tournées en maximisant le nombre de points de livraison visités, sachant que le béton ne peut être déversé par terre (salissure, durcissement...).

À ces deux questions, BÉTON DIRECT a trouvé une réponse unique : utiliser des bacs à béton.

L'idée est de proposer aux clients de recevoir leur produit dans des bacs spécialement conçus pour contenir du béton frais (voir figure 1.9).

En effet, livrer le béton dans des bacs permet de :

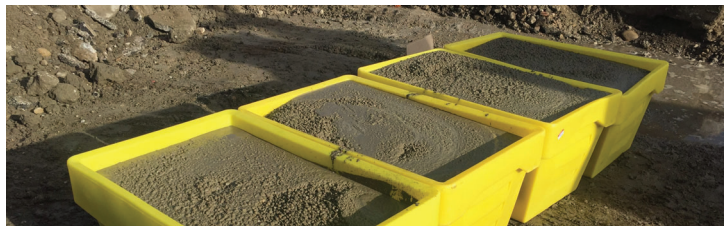


FIGURE 1.9 – 4 Bacs à béton pleins

- s'assurer que le volume reçu par un client correspond bien à ce qu'il a commandé

(les bacs sont gradués) ;

- régler le problème du temps d’immobilisation de la toupie à chaque livraison. Le béton étant livré dans des bacs, la toupie n’est plus obligée d’attendre la fin du coulage pour poursuivre son parcours. Le client récupère ses bacs, et peut piocher dedans à son rythme, tout au long de la journée.

Ce nouveau modèle proposé par BÉTON DIRECT pour livrer du BPE requiert cependant un travail de recherche et de développement sur trois volets distincts :

la conception du bac à béton : le bac utilisé doit satisfaire à certaines exigences en termes de maniabilité, de transportabilité, de lavabilité, de résistance...

la formulation du béton : comme expliqué plus haut, l’idée est de fournir au client des bacs dans lesquels il pourra piocher son béton durant toute la journée, et au fur et à mesure de l’avancement de ses travaux. Pour ce faire, le produit livré doit être utilisable pendant 24h, ce qui n’est pas le cas des bétons "classiques" qui durcissent très vite après fabrication. Il y a donc un travail de préparation de formules de béton spécifiques, qui seraient compatibles avec le mode de livraison en bacs, en ce sens qu’elles permettraient un durcissement retardé du matériau, le rendant utilisable bien plus longtemps ;

l’optimisation des tournées de livraison : afin que le modèle soit rentable, et pour qu’il atteigne les objectifs visés par BÉTON DIRECT, à savoir :

- proposer un service adapté aux petits consommateurs, qui soit, autant que faire se peut, simple, pratique, et peu coûteux ;
- réduire les coûts de la logistique (transport et manutention) ;
- réduire l’impact environnemental lié à la livraison de petites quantités.

Il est nécessaire d’avoir un système d’aide à la décision capable de générer des tournées efficaces, en prenant en compte un certain nombre de contraintes inhérentes au matériau à livrer, et aux véhicules effectuant les tournées.

C’est dans ce contexte qu’est né le besoin d’une collaboration scientifique entre BÉTON DIRECT et le Laboratoire d’Informatique en Images et Systèmes d’Information (LIRIS), afin de mettre au œuvre ce système de gestion et de coordination de tournées de véhicules pour la livraison de béton en bacs.

Le premier travail effectué a été d’imaginer le schéma logistique pouvant répondre aux besoins de l’entreprise. Par schéma logistique, on entend, le choix des types de véhicules

engagés dans les tournées, ainsi que le scénario (ou le déroulement), d'une tournée impliquant ces véhicules, avec tout ce que cela peut engendrer comme contraintes, difficultés, avantages et inconvénients...

Ainsi, deux schémas logistiques ont été étudié :

- le premier fait appel à un unique type de véhicules, le camion grue (voir photo 1.10). Il s'agit d'un véhicule lourd, équipé d'une benne capable d'accueillir une charge conséquente, et d'un bras articulé pour décharger le contenu de la benne. L'idée ici est de charger des bacs à béton pleins dans ce genre de camions, et de les décharger chez les clients à l'aide du bras articulé. Le camion peut également, durant sa tournée, passer chez les clients de la veille pour récupérer les bacs qu'ils auront fini d'utiliser ;



FIGURE 1.10 – Camion grue

- le second schéma logistique implique quant à lui deux types de véhicules : un véhicule utilitaire léger (voir photo 1.11), et un camion toupie (voir photo 1.1).



FIGURE 1.11 – Véhicule utilitaire

Le principe ici est de séparer la livraison des bacs de celle du béton. Ainsi, l'utilitaire

se charge de transporter les bacs **vides** (et donc légers) chez les clients, tandis que la toupie livre le produit frais dans la foulée chez ces mêmes clients. Les bacs vides reçus par les clients servent alors de réceptacles pour accueillir le béton qui arrive chez eux dans la foulée.

Chacun des deux schémas comporte des avantages et des inconvénients. Nous les présentons plus en détail, et séparément, dans les chapitres 2 et 3.

Conclusion

BÉTON DIRECT est une jeune startup innovante qui s'est donné pour mission de simplifier le parcours d'achat de béton prêt à l'emploi à sa clientèle cible, les particuliers et les petits artisans, qui n'ont besoin que de petites quantités de béton.

Pour ce faire, l'entreprise a identifié un certain nombre de leviers pour améliorer l'expérience d'achat de ses clients, tout en leur proposant les meilleurs prix possibles. Parmi ces leviers se trouve la proposition d'un modèle logistique plus adapté à la livraison de béton en petites quantités. En effet, la logistique adoptée jusqu'à présent pour livrer du béton frais est la même, qu'il s'agisse de petites ou de grandes quantités livrées, ce qui est absurde dans la mesure où le matériel utilisé pour transporter des volumes conséquents de BPE s'avère disproportionné et coûteux lorsqu'il est utilisé pour de petites livraisons, sans parler de l'impact environnemental très négatif.

La solution imaginée pour répondre à cette problématique est la mutualisation des moyens de transport à travers la mise en place de tournées permettant d'optimiser le rendement des véhicules, de réduire les coûts de transport, et de minimiser l'impact carbone des livraisons.

Cette réflexion a incité l'entreprise à solliciter une collaboration avec un laboratoire scientifique (Le LIRIS) pour mettre au point un système d'aide à la décision capable de générer et coordonner ces tournées, qui doivent par ailleurs respecter un certain nombre de contraintes liées à la nature du produit transporté, et à celle des véhicules impliqués.

Schéma logistique 1 : Tournées de livraison et de collecte à véhicule unique

Introduction

Afin de proposer la meilleure solution de livraison de béton frais pour les petits consommateurs, qui soit la plus simple et pratique pour les clients, la plus respectueuse pour l'environnement, et la plus rentable pour l'entreprise, deux scénarios de livraison (deux schémas logistiques) ont été étudiés.

Nous nous penchons ici sur le premier schéma logistique proposé, à savoir, la livraison de bacs remplis de béton, puis leur collecte une fois le béton consommé. Nous présentons dans ce chapitre :

- le fonctionnement, les contraintes, et les objectifs de ce premier scénario ;
- un état de l'art sur les travaux de recherche traitant de problématiques proches de celles induites par ce premier scénario ;
- notre approche pour mettre en œuvre un système d'aide à la décision capable de générer des tournées de véhicules efficaces suivant ce schéma logistique ;
- les résultats des tests effectués sur les algorithmes proposés.

2.1 Description du schéma logistique

Le premier scénario de livraison étudié ne fait appel qu'à un seul type de camions. Il est détaillé ci-dessous, du point de vue du client d'abord, puis du point de vue de BÉTON DIRECT.

2.1.1 Point de vue client

Pour le client, le premier scénario étudié comporte les étapes suivantes :

1. le client, dont le besoin en béton ne dépasse pas les $2m^3$, passe commande chez BÉTON DIRECT, en indiquant l'adresse du chantier et la date de livraison souhaitée ;
2. le jour de la livraison, un camion se présente à l'adresse du chantier pour y déposer un ou plusieurs bacs remplis de béton et s'en va immédiatement (le nombre de bacs déposés dépend du volume commandé par le client, sachant qu'un bac peut contenir jusqu'à 500 litres de BPE) ;
3. une fois les bacs reçus, le client a toute la journée pour faire ses travaux en piochant du béton frais, à chaque fois qu'il en a besoin, dans les bacs qui lui ont été livrés ;
4. le lendemain, un camion repasse sur le chantier pour récupérer les bacs livrés la veille, et dont le béton aura été consommé par le client (voir figure 2.1).

2.1.2 Point de vue Béton Direct

Pour BÉTON DIRECT, les événements s'enchaînent de la manière suivante :

1. un camion quitte le dépôt et entame sa tournée chargé de bacs remplis de béton frais, destinés à être livrés à un ensemble de clients ayant passé commande quelques jours auparavant ;
2. au cours de sa tournée, le camion suit un itinéraire lui permettant de passer chez chacun des destinataires des bacs qu'il transporte, ainsi que chez des clients qui ont préalablement (par exemple, la veille) reçu des bacs, qu'ils ont vidé en consommant le béton qu'ils contenaient, et qu'ils doivent donc rendre à BÉTON DIRECT.
3. le camion rentre au dépôt, chargé des bacs vides qu'il a récupéré chez les clients de la veille, et en ayant livré tous les bacs pleins qu'il transportait au début de sa

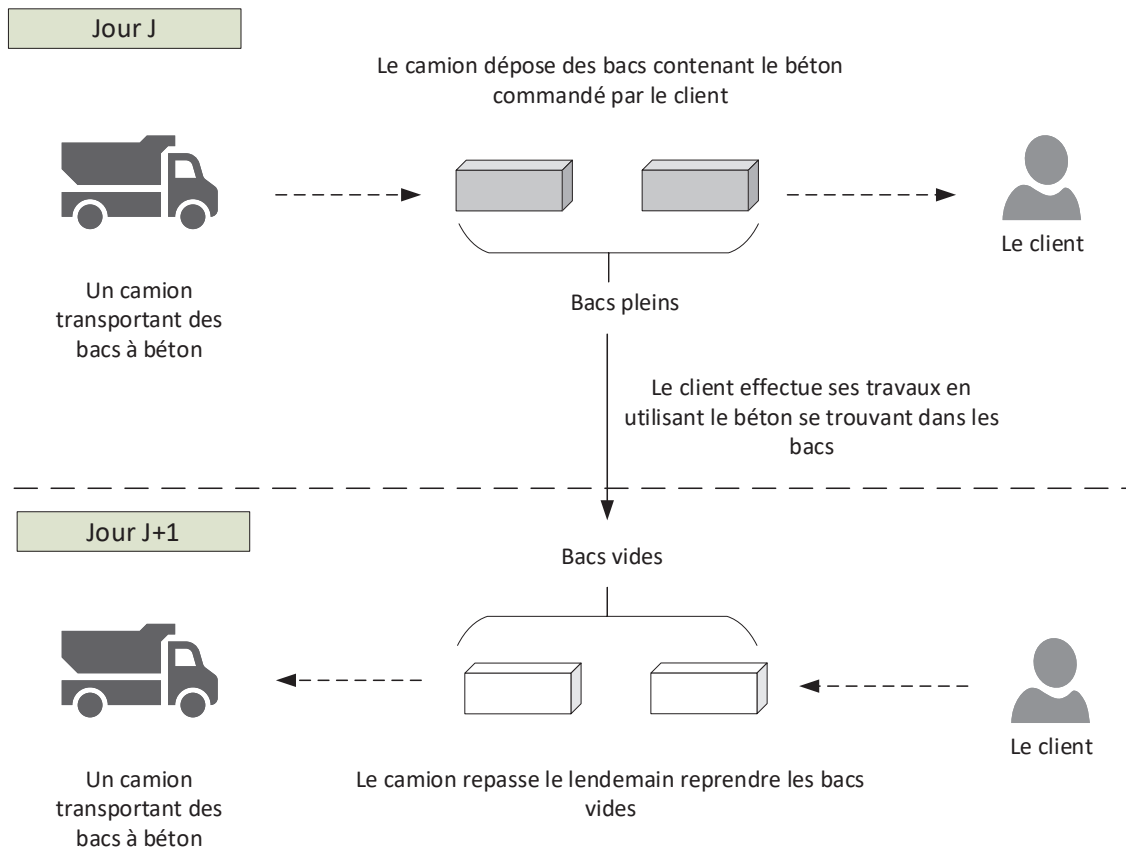


FIGURE 2.1 – Le schéma logistique n° 1 vu par le client

tournée (voir figure 2.2).

2.1.3 Recyclage du béton non consommé

a) Problème

Dans le scénario précédemment décrit, le véhicule effectue sa tournée en alternant livraisons de bacs pleins et collectes de bacs vides. Néanmoins, en pratique, les bacs collectés chez les clients livrés la veille peuvent ne pas être vides. En effet, il arrive régulièrement que des clients surestiment leurs besoins en béton et commandent par conséquent des quantités supérieures à ce dont ils ont réellement besoin pour accomplir leurs travaux. Dans ces cas-là, le béton non consommé reste dans les bacs, où il finit par durcir et se transformer en bloc compact.

Le client dans une telle situation se retrouve alors démuné et impuissant devant le poids considérable du produit durci, et ne peut s'en débarrasser sans l'aide de matériels spécifiques qu'il ne possède pas. De plus, quand bien même il arriverait à déplacer les bacs

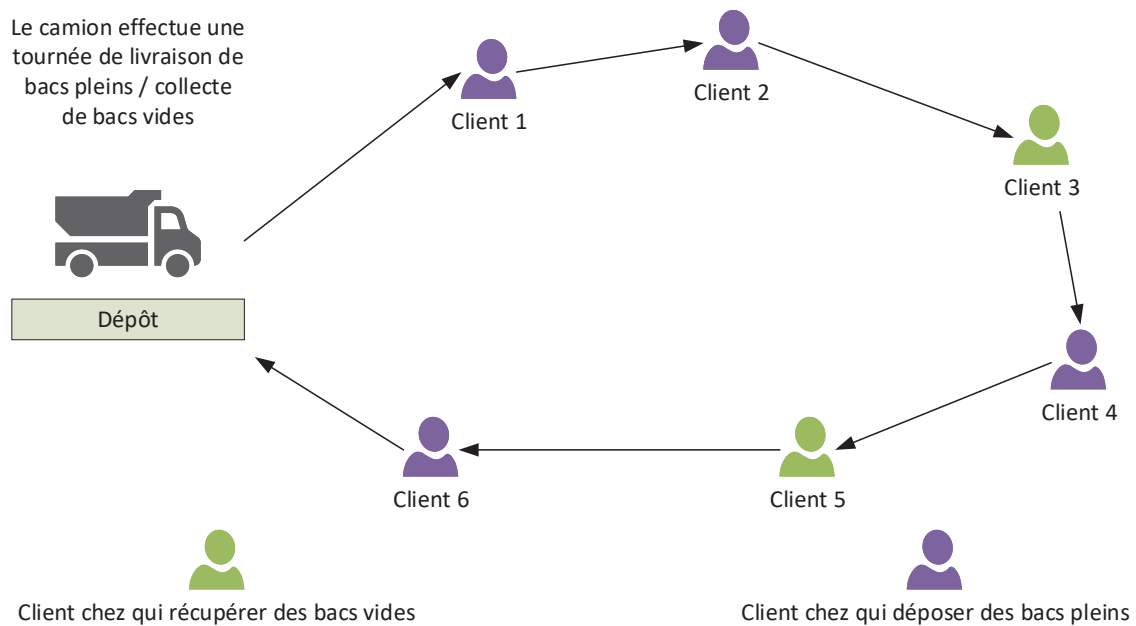


FIGURE 2.2 – Le schéma logistique n° 1 vu par BÉTON DIRECT

contenant le béton non consommé, il ne saurait pas quoi en faire, si ce n'est jeter le produit dans une déchèterie, ou pire, quelque part dans la nature, c'est qui n'est pas souhaitable d'un point de vue écologique.

Le réduction de l'impact environnemental étant un des objectifs qui ont motivé le développement d'un nouveau mode de livraison de BPE en petites quantités, il serait absurde et contre productif d'ignorer la question du béton non consommé. De la même manière, il serait absurde et contre productif, commercialement, d'ignorer cette question. En effet, il est inimaginable de demander à un client de se débarrasser lui-même d'un produit encombrant et très lourd, sans même lui donner les moyens nécessaires pour le faire. Un tel fonctionnement serait très contraignant et ferait sans doute fuir des clients, qui, sans ce problème de traitement des volumes résiduels, pourraient être attirés par la livraison en bacs.

b) Solution

Pour répondre à ce problème, BÉTON DIRECT propose d'exploiter son réseau de partenaires producteurs. En effet, les producteurs de béton possèdent des centrales où le béton est fabriqué. Ces centrales disposent de moyens permettant le recyclage du béton

durci. Ainsi, elles pourraient recevoir les résidus de béton présents dans les bacs, et les recycler pour d'autres usages.

Pour ce faire, le camion qui effectue la tournée de livraison/collecte de bacs doit faire un détour dans une des centrales à proximité pour y déposer le béton non consommé, et ce à chaque fois qu'il récupère des bacs non vides chez un client.

Remarque 1. *Afin de lever toute ambiguïté, on parlera désormais de "centre de recyclage" pour désigner le lieu où doit être acheminé le béton non utilisé pour y être recyclé, le terme "centrale à béton" désignera uniquement le lieu de fabrication du béton.*

En procédant ainsi :

- le consommateur n'a plus à se soucier du devenir du béton qu'il n'a pas consommé ;
- les centrales récupèrent un produit qu'elles peuvent recycler et réutiliser, ce qui leur est profitable ;
- l'impact environnemental est réduit grâce au recyclage.

En résumé, le véhicule a un itinéraire précis à suivre pendant sa tournée, intégrant tous les points de livraison et points de collecte de bacs. Néanmoins, en cours de route, il peut être amené à effectuer un ou plusieurs détours dans un des centres de recyclage à proximité pour y acheminer du béton non utilisé. Ainsi, du point de vue du chauffeur du camion, le processus suivant est à répéter jusqu'à ce que tous les points de passage de la tournée aient été visités (voir figure 2.3) :

1. se diriger vers le prochain point de livraison ou de collecte indiqué sur la feuille de route ;
2. une fois sur place :
 - s'il s'agit d'un point de livraison, déposer le nombre de bacs pleins indiqué, et revenir à l'étape (1) ;
 - s'il s'agit d'un point de collecte, récupérer les bacs et :
 - si les bacs collectés sont tous vides, revenir à l'étape (1) ;
 - si au moins un des bacs récupérés contient du béton non consommé, faire un détour dans un centre de recyclage, y laisser les résidus de béton, et revenir à l'étape (1) ;

La figure 2.4 illustre le déroulement d'une tournée de livraison et de collecte de bacs, avec détours en centre de recyclage.

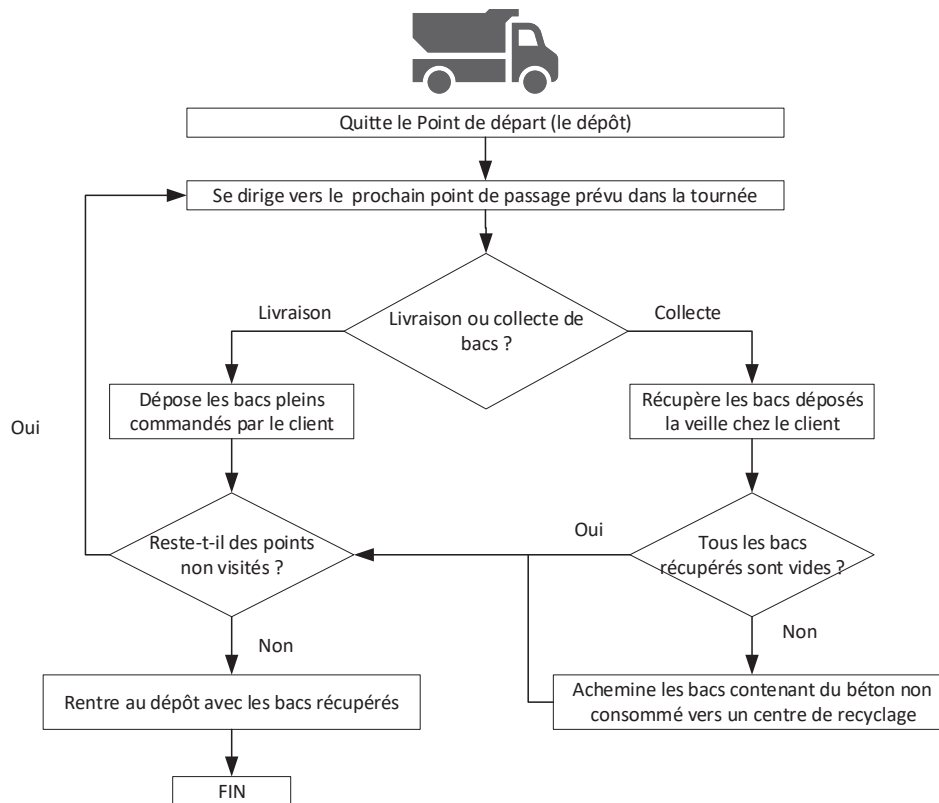


FIGURE 2.3 – Le schéma logistique n° 1 vu par le véhicule effectuant la tournée

2.2 Sujets de recherche sous-jacents

La mise en œuvre d'un système d'aide à la décision capable de générer des tournées de camions efficaces, et suivant le schéma logistique décrit ci-dessus implique de se pencher sur une famille de problèmes très connue et très largement étudiée dans la littérature, les problèmes de tournées de véhicules (ou *Vehicle Routing Problems (VRP)*).

En effet, ce type de problème a, depuis très longtemps, attiré les chercheurs du fait de ses très nombreuses applications dans des domaines divers et variés. Un grand nombre de variantes et de déclinaisons ont été considérées, prenant en compte toutes sortes de variables, de contraintes, et d'objectifs.

Pour ce qui est des tournées évoquées plus haut, un véhicule doit visiter n clients pour livrer ou collecter des bacs à béton. Il s'agit-là d'une variante du problème de tournées de véhicules classique dans laquelle deux opérations distinctes sont possibles à chaque point de passage du véhicule, on parle de problème de tournées de véhicule avec collectes & livraisons (ou *Vehicle Routing Problem with Pickup & Delivery (VRPPD)*).

D'autre part, le schéma de tournées considéré prévoit des passages en centre de recyclage

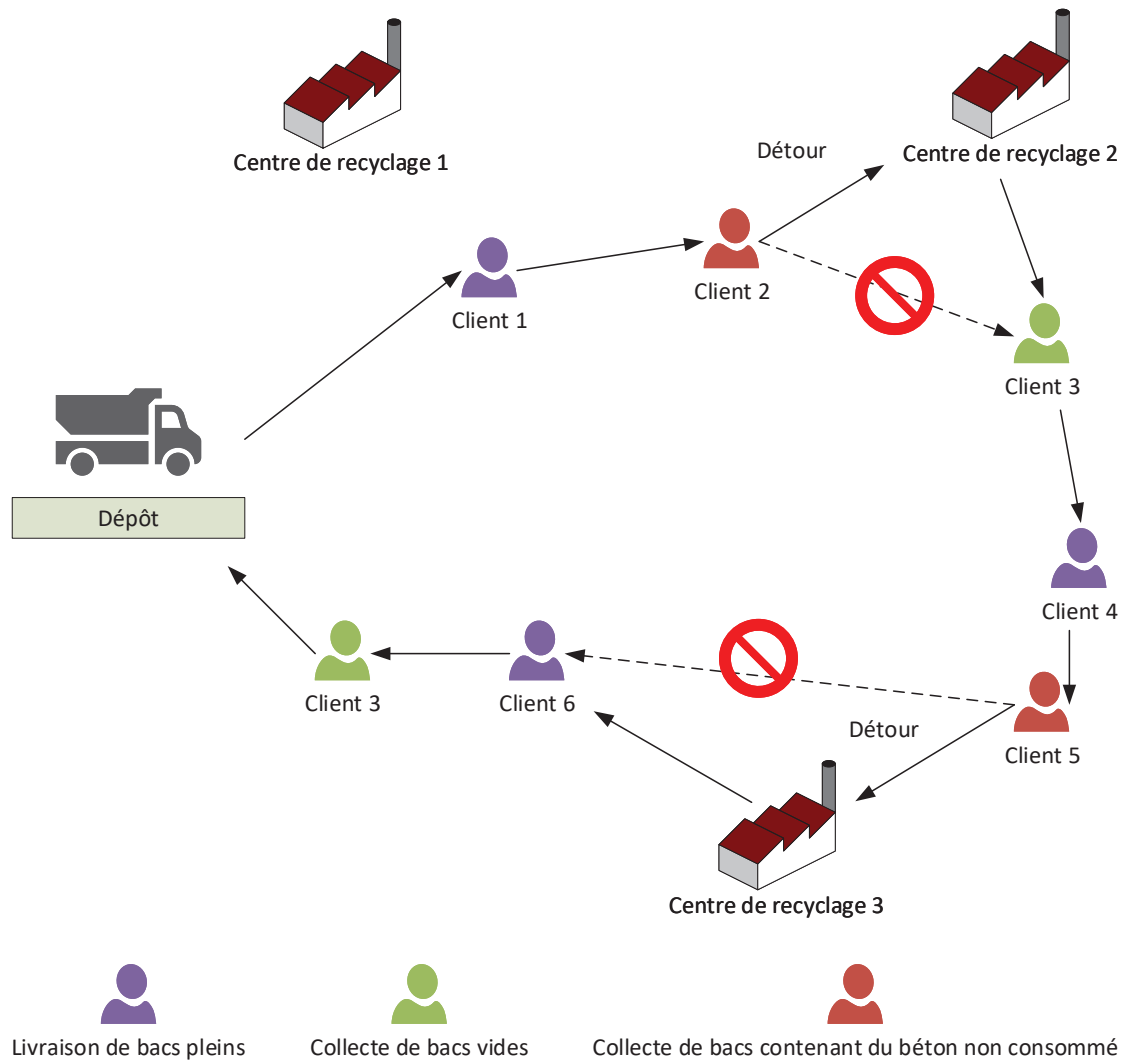


FIGURE 2.4 – Tournée de livraison et de collecte de bacs avec détours en centres de recyclage

à chaque fois que des bacs sont récupérés non vides (à chaque fois que du béton non consommé est récupéré chez un client). Cependant, l'état des bacs à récupérer n'est connu qu'au moment où le véhicule arrive chez le client, et que le chauffeur constate la présence, ou l'absence, de béton dans les bacs qu'il doit collecter. Ce n'est qu'à ce moment-là qu'est prise la décision de poursuivre la tournée normalement, ou de faire un détour en centre de recyclage. Par conséquent, le nombre de détours à effectuer au cours d'une tournée, ainsi que les endroits auxquels ils surviennent sont des informations dont on ne dispose pas au moment où le camion quitte le dépôt pour entamer sa tournée. Les détours sont non planifiables, ce qui rend les tournées sujettes à des changements en cours de route. On parle dans ces cas-là de tournées de véhicules stochastiques ou dynamiques.

Nous nous intéressons ci-dessous aux deux sujets soulevés par le premier schéma logistique

étudié, à savoir, les tournées de véhicules avec collectes & livraisons et les tournées de véhicules stochastiques/dynamiques, et à la manière dont ils sont abordés et traités dans la littérature.

2.3 Tournées de véhicules avec collectes & livraisons

Les problèmes de tournées de véhicules sont une famille de problèmes d'optimisation où il est question de calculer, de façon efficace, le meilleur itinéraire possible pour un ou plusieurs véhicules devant visiter un certain nombre de points, en respectant un certain nombre de contraintes. Cette famille de problèmes peut être scindée en plusieurs classes, parmi lesquelles on trouve les problèmes de tournées de véhicules avec collectes & livraisons.

Pour cette classe de problèmes, les véhicules impliqués dans une tournée ne sont plus uniquement associés à un ensemble de clients à visiter, mais également à des opérations à effectuer chez chacun de ces clients. Ainsi, on distingue deux types d'opérations possibles, la livraison et la collecte. Le véhicule effectuant une tournée doit, selon le besoin du client, lui livrer un ou plusieurs biens provenant d'un ou plusieurs nœuds sources (qui peuvent être le dépôt, ou d'autres clients), et/ou collecter des biens qui seront acheminés vers des nœuds destinations (qui peuvent, là encore, être le dépôt, ou d'autres clients).

2.3.1 Classification

[Berbeglia et al., 2007] proposent une classification des *Pickup & Delivery problems* (PDP) basée sur les trois critères suivants :

La configuration Il s'agit de la manière dont sont reliés les nœuds requérant une livraison et ceux requérant une collecte. On distingue 3 structures principales pour les PDP :

Many-to-Many (M-M) : Dans cette configuration, les biens provenant des clients requérant une collecte peuvent être livrés à n'importe quel client requérant une livraison. En d'autres termes, si on considère les clients requérant une collecte comme des nœuds sources, et ceux requérant une livraison comme des nœuds destinations, alors, les biens peuvent être acheminés de n'importe quel nœud source,

à n'importe quel nœud destination.

One-to-Many-to-One (1-M-1) : Ici, l'ensemble de biens à livrer se trouve au niveau du dépôt, et l'ensemble des biens à collecter se trouve chez les clients. Le principe est alors d'effectuer une tournée pour transporter les biens à livrer du dépôt vers les clients, tout en collectant des biens chez les clients, vers le dépôt. La tournée du laitier, où un véhicule livre des bouteilles de lait pleines tout en récupérant des bouteilles vides est un exemple de cette configuration.

One-to-One (1-1) : Chaque bien considéré ici a une unique source et une unique destination possibles. Le transport de courrier est un exemple célèbre de cette configuration.

Le déroulement des visites Ce critère indique comment doivent être effectuées les opérations de collecte et de livraison à chaque nœud visité. Ainsi, plusieurs scénarios sont possibles. Dans certains problèmes, un véhicule doit visiter chaque nœud une seule fois pour y effectuer, simultanément, des opérations de livraison et de collecte. Un autre scénario possible serait d'effectuer pour chaque nœud visité, les deux opérations, mais en se donnant la possibilité de le faire en deux temps (en deux visites). Pour certains problèmes, un client ne peut requérir qu'une livraison ou une collecte, mais pas les deux...

Le nombre de véhicules Il s'agit du nombre de véhicules considérés dans la tournée. On trouve dans la littérature des problèmes à 1 ou plusieurs véhicules.

Nous présentons dans ce qui suit quelques exemples de PDP abordés dans la littérature.

2.3.2 Revue de la littérature

a) Problèmes *Many-to-Many* (M-M)

Le *One-commodity Pickup & Delivery Traveling Salesman Problem (1-PDTSP)* est une variante de PDP intéressante et assez peu étudiée dans la littérature. Le problème consiste, pour un véhicule unique, à transporter un unique type de biens, entre des nœuds sources, et des nœuds destinations. Le véhicule est donc amené à charger et à décharger les biens de sorte à satisfaire toutes les demandes. À ces hypothèses, s'ajoute également une contrainte qui impose une capacité de chargement maximale au véhicule durant sa

ournée. Cette contrainte complexifie davantage le problème, qui est non seulement *NP-Difficile*, mais dont le simple fait de déterminer si une solution est faisable ou non est un problème *NP-Complet* [Hernández-Pérez, 2004].

Si l'on suit la classification suivant les trois critères présentés plus haut, on peut considérer que le 1-PDTSP :

- définit une structure *Many-to-Many*, car il s'agit d'acheminer des biens entre des nœuds sources et des nœuds destinations, sans affecter, au préalable, une source spécifique, à une destination en particulier ;
- permet une livraison ou une collecte à chaque point visité, mais pas les deux ;
- fait appel à un unique véhicule.

Le problème a été introduit par [Hernández-Pérez and Salazar-González, 2004a]. Dans ce travail, un algorithme basé sur un *Branch & Cut* a été étudié pour traiter des instances de petite taille (jusqu'à 40 nœuds visités). Plus tard, [Hernández-Pérez and Salazar-González, 2004b] proposeront deux heuristiques pour traiter des instances plus grandes (jusqu'à 500 points visités). Ce papier propose notamment une adaptation de l'algorithme du plus proche voisin utilisé pour le TSP classique, visant à privilégier l'alternance entre livraisons et collectes, afin de maximiser les chances d'obtenir des solutions faisables. Les algorithmes 2-opt et 3-opt [Lin, 1965, Johnson and McGeoch, 1997], largement utilisés pour générer des solutions voisines d'une solution donnée, ont également été modifiés pour tenir compte du risque d'infaisabilité des solutions voisines explorées.

De la même manière que le *1-PDTSP*, le problème de *Swapping* est une autre variante de PDP qui définit également une structure *Many-to-Many*, mais qui contrairement au *1-PDTSP*, implique plusieurs véhicules, plusieurs types de biens à transporter, et plusieurs opérations à effectuer pour chaque point visité (une collecte **et** une livraison).

Pour ce problème, introduit par [Anily and Hassin, 1992], il s'agit de procéder à des échanges de biens entre les nœuds visités. Ici, les ensembles représentant, respectivement, les nœuds sources et les nœuds destinations, ne sont pas disjoints. Mieux encore, chaque nœud visité est à la fois source et destination pour **deux** types de biens. Ainsi, si on considère trois types de biens x , y , et z , chaque nœud à visiter possède, avant le début de la tournée, une unité du bien x , ou y , ou z , et veut l'échanger contre une unité d'un bien qu'il ne possède pas. Chaque nœud requiert donc la collecte du bien dont il n'a plus

besoin (qu'il veut échanger), et la livraison du bien qu'il veut en échange. Ce travail de permutation de biens entre nœuds est alors effectué par plusieurs véhicules.

Il est également à noter que ce problème de *Swapping* se décline en deux variantes, la version non-préemptive, et la préemptive. Dans la première, les biens transportés ne peuvent être livrés qu'à leurs destinataires finaux, alors que la seconde variante, la préemptive, autorise les véhicules à déposer des biens à livrer, temporairement, si cela s'avérait plus pratique, chez des nœuds "intermédiaires", avant de repasser les prendre plus tard pour les acheminer vers les nœuds finaux.

Dans leur travail, [Anily and Hassin, 1992] montrent que le problème de *Swapping* est *NP-Difficile*. Ils ont proposé deux heuristiques et ont montré que leurs performances dans le pire cas ne dépassaient pas le ratio de 2.5 par rapport aux performances optimales. Ce ratio sera ramené à 2 par [Chalasanani and Motwani, 1999], qui obtiennent cependant cette borne pour le cas particulier où ne sont considérés que deux types de biens.

Il est également à signaler que le problème de *Swapping* a aussi été étudié sur des cas particuliers de graphes. [Anily et al., 2011] ont par exemple étudié le cas du *Swapping* préemptif sur un arbre, et ont montré que le problème était *NP-Difficile*. Plus généralement, il a été montré que le *Swapping* était un problème *NP-Difficile* dans les cas préemptif, non-préemptif et mixte, pour les graphes généraux [Anily and Hassin, 1992], ainsi que pour les arbres [Frederickson and Guan, 1993, Anily et al., 2011]. Le problème est cependant polynomial pour les graphes en forme de ligne [Anily et al., 1999].

b) Problèmes *One-to-One* (1-1)

Les problèmes présentés ci-dessus définissent une structure *Many-to-Many*. Cette configuration est moins étudiée que les deux autres (*One-to-One* et *One-to-Many-to-One*), notamment parce qu'elle comporte moins d'applications concrètes dans l'industrie.

Les problèmes de type *One-to-One* sont pour leur part beaucoup plus fréquemment rencontrés dans la vie quotidienne. Un des plus célèbres d'entre eux est désigné dans la littérature sous le nom de *Vehicle Routing Problem with Pickups & Deliveries* (VRPPD).

Globalement, il s'agit, pour un ensemble de véhicules, de satisfaire les demandes de clients souhaitant envoyer des biens qu'ils ont en leur possession, vers des destinations de leur choix. Un client X qui souhaite envoyer des biens est alors considéré comme un nœud source chez qui un véhicule devra passer pour collecter les biens à envoyer, et la destination

choisie par le client X est considérée comme un nœud destination requérant la livraison de la totalité des biens collectés chez le client X . Ce problème, ancien et universel, se pose notamment pour les services d'envoi de courrier.

Les VRPPD se déclinent eux aussi sous diverses formes. Le *Single Vehicle Routing Problem with Pickups & Deliveries* (SVRPPD) est une variante où un unique véhicule est considéré dans les tournées. Le problème a été étudié avec ([Van der Bruggen et al., 1993, Desrosiers et al., 1986]) et sans ([Stein, 1978, Psaraftis, 1983]) contrainte de capacité sur le véhicule. Il est également souvent associé à des contraintes imposant des fenêtres de temps pour livrer les biens ([Desrosiers et al., 1986, Van der Bruggen et al., 1993])...

Diverses méthodes approchées et exactes ont été proposées pour aborder le problème. Pour les méthodes approchées, des algorithmes constructifs ([Stein, 1978]) et itératifs (faisant appel à des recherches locales) ont été considérés. Citons par exemple [Healy and Moll, 1995] qui proposent une adaptation de l'algorithme *2-opt* [Croes, 1958] qui minimise le risque de tomber sur des solutions infaisables durant la recherche locale, ou encore, [Van der Bruggen et al., 1993] dont l'approche, proposée pour un SVRPPD avec contrainte de capacité et fenêtres de temps, a obtenu des résultats proches de l'optimum pour des instances allant jusqu'à 50 nœuds visités.

Plus récemment, une nouvelle variante a fait l'objet de nombreux travaux. Elle introduit une contrainte imposant que les derniers biens collectés soient les premiers à être livrés. En effet, pour certaines applications dans l'industrie, il arrive que le type de véhicule, ou que la nature des biens transportés, imposent de devoir empiler les objets collectés pendant leur acheminement, rendant difficile, voire impossible, la livraison des biens qui se trouvent en bas de la pile, et obligeant ainsi à recourir à la stratégie "dernier arrivé, premier servi" (LIFO). Parmi les travaux menés sur ce sujet, [Carrabs et al., 2007] proposent une méthode approchée en introduisant trois nouveaux mécanismes d'exploration de voisinage, adaptés au problème, qu'ils intègrent dans une recherche locale, tandis que [Cordeau, 2006] se penche sur la résolution exacte du problème, en proposant notamment trois programmes linéaires en nombres entiers, testés sur des instances allant jusqu'à 25 nœuds visités.

Lorsque un VRPPD ne traite pas de transport de biens, mais de personnes, on parle de *Dial a Ride Problem* (DARP). Il s'agit, là encore, d'un problème largement étudié dans

la littérature au vu des nombreuses applications concrètes qui y sont associées, comme le transport de personnes en situation de handicap. Ainsi, certains travaux présentent des approches ayant fait l'objet de tests "grandeur nature" dans de nombreuses villes, notamment européennes (Copenhague [Madsen et al., 1995], Berlin [Borndörfer et al., 1999], Bruxelles [Rekiek et al., 2006]...).

Deux des contraintes les plus souvent associées aux DARP sont la contrainte de capacité sur les véhicules transportant les personnes, et les fenêtres de temps définissant les horaires auxquels les personnes doivent être prises en charge. Quant aux objectifs visés, on trouve souvent, en plus des objectifs liés à la réduction des coûts de transport, ceux en rapport avec le caractère agréable (ou désagréable) du service fourni aux personnes. Typiquement, les travaux s'intéressent à minimiser le temps d'attente des personnes avant leur prise en charge, ainsi que la durée des trajets effectués.

Ainsi, [Psaraftis, 1980] étudient un problème où des personnes souhaitent être prises en charge le plus tôt possible, par un unique véhicule disponible. Ici, il n'y a pas de fenêtres de temps imposées au véhicule pour prendre en charge les différentes personnes, mais il s'agira de minimiser une fonction objectif représentant la somme pondérée du temps de service total du véhicule (ou, en d'autres termes, les coûts associés au service), et du niveau d'insatisfaction des personnes. Ce dernier dépend de deux paramètres : le temps d'attente des personnes avant leur prise en charge par le véhicule et le temps passé dans le véhicule. Les auteurs résolvent ce problème de façon exacte, uniquement pour de petites instances, à l'aide d'une approche basée sur la programmation dynamique.

Un autre type de contraintes est étudié par [Jaw et al., 1986]. Les auteurs se penchent sur un problème impliquant plusieurs véhicules, et où chaque personne transportée choisit l'heure de départ, ou l'heure d'arrivée souhaitée (mais pas les deux). Un véhicule ne peut alors prendre en charge une personne qu'après l'heure de départ choisie, et doit arriver à destination avant l'heure d'arrivée désirée. De plus, les véhicules ne sont pas autorisés à s'arrêter au cours d'un trajet. Partant de ces contraintes, l'objectif est de minimiser les coûts de transports, tout en maximisant la qualité de service. Celle-ci est par ailleurs définie par les durées des trajets effectués, et le respect des heures de départs et d'arrivées choisies par chacune de personnes transportées. Un algorithme constructif a été développé pour résoudre ce problème. Le principe est de classer les personnes selon l'heure de départ souhaitée, de la plus proche, à la plus éloignée, puis d'affecter les personnes ayant

choisi les heures de départ les plus proches aux véhicules disponibles. Par la suite, et à chaque itération, l'algorithme essaye d'affecter la prochaine personne non encore prise en charge dans la liste ordonnée, à chacun des véhicules disponibles. Le véhicule pour qui le déplacement chez la personne est le moins coûteux est choisi pour la prendre en charge. L'algorithme a été testé sur des données impliquant plus de 2600 personnes, et jusqu'à 28 véhicules.

c) Problèmes *One-to-Many-to-One* (1-M-1)

Les quelques VRPPD présentés ci-dessus, qui définissent une structure *One-to-One*, sont clairement plus abordés dans les travaux de recherche que les problèmes *Many-to-Many* (dont quelques exemples ont été exposés plus haut dans cette section). Cela est dû au caractère plus pratique et plus concret des problèmes *One-to-One*, qui sont omniprésents dans des domaines divers et variés.

De la même manière, Si l'on devait classer les familles de PDP selon l'abondance des travaux de recherche qui y sont consacrés, les problèmes de type *One-to-Many-to-One* seraient probablement mieux positionnés que les *Many-to-Many* au vu de la variété d'applications réelles qu'ils offrent.

Ainsi, pour les problèmes *One-to-Many-to-One* (1-M-1), il s'agit, pour un ou plusieurs véhicules, de transporter des biens d'un dépôt vers un ensemble de nœuds, tout en collectant, chez ces mêmes nœuds, des biens pour les acheminer vers le dépôt. La logistique inverse constitue le principal champ d'application de ce genre de problèmes. Cela couvre notamment des problèmes du type "tournée du laitier", où des véhicules doivent livrer des bouteilles de lait pleines tout en collectant des bouteilles vides précédemment livrées.

Il existe deux déclinaisons pour les problèmes 1-M-1. Dans la première, au moins un des nœuds visités requiert une livraison **et** une collecte. Dans la seconde, tous les nœuds visités requièrent une livraison ou une collecte, mais pas les deux.

Lorsque certains nœuds visités requièrent une livraison et une collecte, plusieurs types de chemins peuvent être envisagés pour les véhicules effectuant les tournées. En effet, [Gribkovskaia et al., 2007] recensent quatre formes de parcours différentes :

Si chaque nœud doit être visité exactement une fois durant une tournée,
alors les véhicules se retrouvent obligés d'effectuer les opérations de livraison et de collecte simultanément pour chaque nœud où cela est nécessaire. On parle de **par-**

cours Hamiltonien.

Si les nœuds peuvent être visités à deux reprises durant une tournée, alors, trois choix s'offrent aux véhicules :

1. Visiter tous les nœuds une première fois en effectuant des livraisons (toutes les livraisons sont alors accomplies lorsque le véhicule arrive au bout du chemin "aller"). Puis refaire le chemin à l'envers (chemin "retour") pour effectuer les collectes. Dans ce cas, le dernier nœud visité sur le chemin "aller" est également le premier visité sur le chemin "retour". C'est également le seul nœud chez qui les opérations de livraison et de collecte sont effectuées simultanément. Ainsi, en ayant cinq nœuds à visiter, un itinéraire possible serait $1- > 2- > 3- > 4- > 5- > 4- > 3- > 2- > 1$. On parle alors de **parcours "double chemin"** (ou *double path*). Ce type de parcours présente l'avantage de ne pas générer de solutions infaisables lorsque les véhicules sont soumis à des contraintes de capacité de chargement. En effet, puisque toutes les livraisons se font avant les collectes, aucune violation de la contrainte de capacité ne peut survenir durant une tournée (pour peu que le véhicule soit en capacité de transporter tous les biens collectés, simultanément) ;
2. Scinder l'ensemble des nœuds à visiter en deux sous-ensembles. Les nœuds du premier sous-ensemble sont d'abord visités en suivant un chemin donné (exemple : $1- > 2- > 3$), pour y effectuer des livraisons, puis les nœuds du deuxième sous-ensemble sont visités (exemple : $4- > 5- > 6$). Des livraisons et collectes simultanées sont alors effectuées lors de ces visites. Enfin, les nœuds du premier sous-ensemble sont visités à nouveau, dans le sens inverse du chemin parcouru lors des livraisons ($3- > 2- > 1$), en vue d'effectuer des collectes. Ce qui nous donne le parcours complet : $1- > 2- > 3- > 4- > 5- > 6- > 3- > 2- > 1$. Les nœuds 4, 5 et 6 forment une boucle qui, associée au chemin formé par les nœuds 1, 2 et 3 dessine un parcours en forme de lasso. D'où le nom de ce type de parcours, le **parcours lasso** ;
3. Visiter chaque nœud, sans qu'il y ait de forme ou de modèle à respecter. Certains nœuds seront visités deux fois quand d'autres ne le seront qu'une seule fois. Ce type de parcours est le plus flexible de tous. On parle de **parcours général**.

[Gribkovskaia et al., 2007] mettent en lumière une inégalité intéressante reliant les quatre

types de parcours décrits ci-dessus. En effet, ils montrent que si, pour une instance donnée, les valeurs des solutions optimales suivant les parcours général, lasso, Hamiltonien, et double chemin sont respectivement notées z_G^* , z_L^* , z_H^* , et z_D^* , alors l'inégalité suivante est vérifiée : $z_G^* \leq z_L^* \leq z_H^* \leq z_D^* \leq 2z_G^*$. Ceci rejoint l'idée intuitive selon laquelle le parcours général offre plus de flexibilité, et donc plus de marge de manœuvre pour obtenir des solutions plus efficaces. Et que le parcours double chemin est le plus contraignant dans le sens où tous les nœuds doivent être visités à deux reprises.

Dès lors qu'un problème 1-M-1 est abordé dans un travail de recherche, un ou plusieurs types de parcours peuvent être considérés. Les méthodes approchées sont alors largement utilisées [Gribkovskaia et al., 2001, Hoff and Løkketangen, 2006, Hoff et al., 2009], et notamment les métaheuristiques. À titre d'exemple, [Hoff and Løkketangen, 2006] proposent une recherche Tabou faisant appel à l'algorithme *2-opt* pour générer des solutions lasso pour un problème à véhicule unique, tandis que [Hoff et al., 2009] proposent également une recherche Tabou pour construire des solutions lasso, mais pour un problème multi-véhicules cette-fois ci. Ils définissent des mécanismes de génération de voisinage adaptés à leur problème. Leur algorithme est paramétrable de sorte à permettre son utilisation pour générer des solutions Hamiltoniennes en plus des solutions lasso pour lesquelles il a été conçu. Des expérimentations impliquant des instances allant jusqu'à 25 véhicules et 260 nœuds visités sont réalisées.

Pour le même type de problèmes, mais pour générer des solutions Hamiltonienne cette fois-ci, les méthodes approchées, dont les métaheuristiques, notamment celles basées sur l'exploration de voisinages, sont également étudiées dans plusieurs travaux. Citons par exemple le travail de [Bianchessi and Righini, 2007] qui comparent plusieurs méthodes constructives, et méthodes itératives (recherches locales et recherches Tabou), avec différentes définitions de la notion de voisinage. La recherche locale à voisinage variable qu'ils proposent obtient des solutions très satisfaisantes d'après les auteurs.

Dans la même lignée, [Montané and Galvao, 2006] proposent également un recherche Tabou pour un problème similaire. Ils définissent le voisinage par des permutations de nœuds entre véhicules, ainsi que par des permutations entre des nœuds associés au même véhicule. Leur approche a été testée sur des instances atteignant les 400 nœuds. Les résultats obtenus sont jugés satisfaisants.

Bien que les méthodes approchées soient majoritaires dans ce qu'on a pu trouver comme

travaux dans la littérature impliquant des problèmes 1-M-1 faisant appel à des solutions Hamiltoniennes, il est à signaler que certaines études se sont penché sur des méthodes de résolution exactes. Ainsi, deux *Branch & Price* ont par exemple été proposés par [Angelelli and Mansini, 2002] et [Dell’Amico et al., 2006]. Le premier a permis de résoudre le problème de façon optimale pour des instances allant jusqu’à 20 nœuds et 5 véhicules, tandis que le second a pu traiter des instances atteignant les 40 nœuds, et dépassant les 20 véhicules.

Comme évoqué précédemment, les problèmes 1-M-1 se scindent en deux groupes, ceux dans lesquels au moins un nœud visité requiert une livraison **et** une collecte, et ceux pour lesquels tous les nœuds visités requièrent une livraison, ou une collecte, mais pas les deux. Nous avons abordé plus haut le cas du premier groupe.

Naturellement, pour le second groupe, puisqu’il n’y a besoin que d’une seule opération (livraison ou collecte) pour chaque nœud visité, les types de parcours général, lasso, et double chemin n’ont plus de sens, car ils impliquent de multiples passages par au moins une partie des nœuds. Ici donc, pour un véhicule donné, seul est pertinent le parcours permettant de passer une et une seule fois par chaque nœud (en d’autres termes, le parcours Hamiltonien).

Ce type de problèmes est abordé de deux façons différentes dans la littérature selon qu’il y ait présence ou non d’une contrainte imposant d’effectuer toutes les livraisons d’abord, avant de passer aux collectes. En effet, cette contrainte rend d’éventuelles contraintes de capacité inutiles car tous les biens à livrer sont d’abord déchargés pour laisser de la place aux biens devant être collectés. Il n’y a donc pas de risque de se retrouver avec un véhicule chargé, à la fois de bien à livrer, et de biens à collecter, ce qui pourrait mener à une violation d’une éventuelle contrainte de capacité. Évidemment, la seule contrainte de capacité à respecter pour les véhicules dans ce cas-là est de disposer d’une capacité suffisante pour transporter tous les biens collectés, simultanément.

Parmi les travaux se penchant sur des 1-M-1 où les livraisons doivent précéder les collectes, [Gendreau et al., 1996] étudient le cas où un unique véhicule est considéré. Ils proposent ainsi six heuristiques différentes, inspirées du travail effectué par [Gendreau et al., 1992]. Les heuristiques ont été testées et comparées sur des instances atteignant les 1000 nœuds, et les meilleures approches ont permis d’obtenir des résultats très proches de l’optimum.

Plus tard, [Mladenović and Hansen, 1997] reprendront le même problème et amélioreront les résultats obtenus par [Gendreau et al., 1996] en enrichissant leurs heuristiques avec une recherche locale à voisinage variable.

Pour le cas où plusieurs véhicules sont considérés, citons [Brandao, 2006] qui décrivent une recherche Tabou qui démarre d'une solution initiale où deux chemins Hamiltoniens (un reliant les nœuds requérant des livraisons, puis un second pour les collectes) sont construits à l'aide de l'heuristique *Nearest Neighbour*, puis reliés entre eux de différentes manières. Cette approche a permis d'obtenir de meilleurs résultats que ceux obtenus par des études antérieures. Signalons également la présence de travaux proposant des approches de résolution exactes pour des 1-M-1 impliquant plusieurs véhicules ainsi qu'une contrainte de précédence entre les livraisons et les collectes. À titre d'exemple, [Yano et al., 1987] développent une approche basée sur un *Branch & Bound* pour résoudre le problème de façon optimale. Cependant, ce genre de travaux restent très limités en termes de tailles d'instances considérées.

Les problèmes 1-M-1 peuvent s'avérer encore plus complexes à aborder lorsqu'il n'y a pas de contrainte de précédence entre livraisons et collectes. En effet, si l'on se donne la possibilité de parcourir les nœuds requérant des livraisons et ceux requérant des collectes dans un ordre quelconque, cela peut donner des solutions plus flexibles, et donc, potentiellement meilleures que lorsqu'on est contraint de prioriser certains nœuds par rapport à d'autres. Néanmoins, si le problème à traiter comporte une contrainte de capacité sur le(s) véhicule(s) impliqué(s) dans les tournées, alors certaines solutions générées pourraient s'avérer infaisables, car ici, contrairement au cas où on effectue toutes les livraisons avant de passer aux collectes, un véhicule peut se retrouver à transporter des biens à livrer en même temps que des biens collectés, d'où le risque, pour le véhicule, de ne pas être en capacité de supporter une telle charge, et d'où la notion de solution infaisable.

Pour aborder ce problème pour le cas d'un véhicule unique, [Mosheiov, 1994] propose une heuristique constructive en deux étapes qui est une extension de l'algorithme *Cheapest Insertion* ([Golden et al., 1980]). La première étape consiste à construire une tournée incluant le dépôt et les nœuds "livraison" à l'aide de l'heuristique *Cheapest Insertion*. Une fois que la tournée est générée, les nœuds "collecte" sont insérés, itérativement, toujours en suivant la logique *Cheapest Insertion*, et en s'assurant que la solution en cours de construction reste faisable après l'insertion de chacun des nœuds. Pour rappel, étant

donné un nœud, l'idée de l'heuristique *Cheapest Insertion* est de l'insérer à l'endroit impliquant la plus petite augmentation du coût total de la tournée en cours de construction. Les résultats obtenus lors des tests sont jugés satisfaisants.

Plus récemment, [Süral and Bookbinder, 2003] reprennent le problème, toujours avec un seul véhicule, mais en ajoutant une nouvelle hypothèse : les collectes deviennent facultatives et non plus obligatoires. En d'autres termes, le véhicule n'est plus contraint de visiter tous les nœuds requérant une collecte. Cependant, chaque collecte effectuée procure un revenu supplémentaire. Ainsi, la fonction objectif considérée par les auteurs est la minimisation du coût de la tournée moins la somme des revenus engrangés grâce aux collectes effectuées. Le problème a été résolu de façon optimale pour des instances allant jusqu'à 30 nœuds à l'aide un programme linéaire en nombres entiers.

De la même manière que [Süral and Bookbinder, 2003], [Alshamrani et al., 2007] reprennent eux aussi le problème 1-M-1 à un véhicule et sans contrainte de précédence entre les opérations de livraison et de collecte, mais en y ajoutant à leur tour une hypothèse supplémentaire : les collectes doivent être effectuées durant des périodes précises, et le non respect de ces périodes engendre des pénalités sous forme de coûts supplémentaires et d'insatisfaction des "clients". L'étude est menée sur un cas réel, celui d'un réseau de distribution de sang rattaché à la croix rouge américaine. La solution proposée fait appel à une heuristique pour générer une solution faisable sur laquelle est appliquée, par la suite, une recherche itérative utilisant l'algorithme *Or-opt* ([Or, 1977]) en vue de l'améliorer.

Pour finir, d'autres travaux se penchent sur le même problème, mais en prenant en considération plusieurs véhicules. Citons par exemple [Wade and Salhi, 2002] qui proposent une heuristique constructive pour générer des tournées et qui, pour minimiser le risque de violations des contraintes de capacité imposées aux véhicules, définissent un paramètre α qui représente le pourcentage de biens déjà livrés par rapport au total des biens à livrer, et interdisent les collectes tant que ce pourcentage n'a pas dépassé un certain seuil fixé à l'avance. Ainsi, pour les cas extrêmes, si $\alpha = 0\%$, aucune restriction n'est imposée lors de la tournée sur la nature du prochain nœud à visiter. À l'inverse, si $\alpha = 100\%$, alors les collectes ne sont possibles qu'une fois toutes les livraisons effectuées (ce qui revient à imposer une contrainte de précédence entre livraisons et collectes).

2.3.3 Synthèse

Les problèmes de tournées de véhicules impliquant des livraisons et des collectes (PDP) constituent une classe importante de problèmes de tournées de véhicules (VRP). Leur omniprésence dans des secteurs nombreux et variés tels que l'industrie, le transport, la logistique, les télécommunications et la santé explique le grand intérêt qui leur est porté par les chercheurs depuis des années.

Les PDP se manifestent souvent de diverses manières, ou sous diverses formes. Selon le contexte considéré, différentes contraintes peuvent être imposées, et différents objectifs peuvent être visés. Pour s'y retrouver au milieu des nombreuses variantes étudiées par les chercheurs depuis des décennies, des classifications ont été proposées. Elles prennent en compte des critères pour distinguer et différencier les nombreuses versions de PDP. Nous avons exposé plus haut la classification proposée par [Berbeglia et al., 2007]. Elle fait appel à 3 critères : la configuration, ou la structure des tournées voulues, le déroulement des visites, et le nombre de véhicules (voir la figure 2.5 et la section 2.3.1 pour plus de détails).

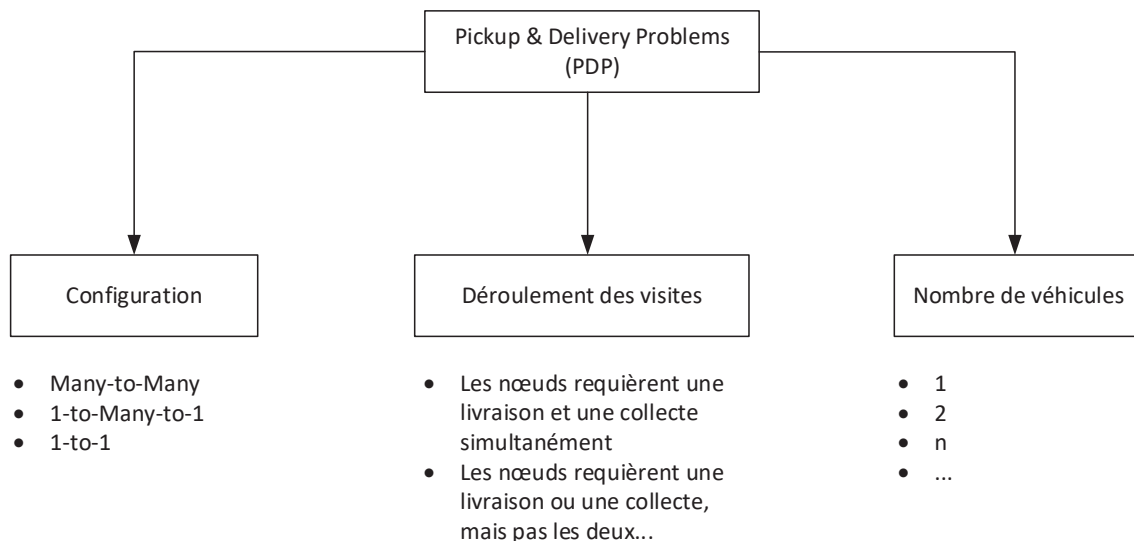


FIGURE 2.5 – Classification des *Pickup & Delivery Problems* (PDP) selon [Berbeglia et al., 2007]

Ainsi, suivant les critères régissant cette classification, nous pouvons considérer que le problème induit par le schéma logistique présenté dans la section 2.1 :

- définit une structure *One-to-Many-to-One*, car il s'agit de livrer des bacs à béton pleins se trouvant dans un dépôt, tout en collectant des bacs vides chez des clients.

Il s'agit donc d'un problème de logistique inverse ;

- autorise une livraison ou une collecte à chaque nœud visité, mais pas les deux ;
- implique la mobilisation d'un unique véhicule.

Ce type de problèmes est *NP-Difficile*. Il est souvent étudié dans la littérature avec des méthodes approchées (heuristiques constructives, métaheuristiques...). Les travaux ayant recours à des méthodes de résolution exactes pour ce type de problèmes sont plus rares et sont limités en termes de taille des instances résolues.

Dans le schéma logistique que nous considérons dans ce chapitre, la contrainte qui impose de détourner le véhicule de son itinéraire pendant les tournées afin d'acheminer les résidus de béton non consommés vers des centres de recyclage implique de devoir construire des tournées en n'ayant pas, à l'avance, toutes les informations sur les nœuds à visiter. En effet, le fait de devoir passer en centre de recyclage à tel ou tel moment est conditionné par la collecte de bacs non vides chez les clients, information dont nous ne disposons pas à l'avance. Le fait de devoir manipuler des données non précises, incomplètes, voire absentes pour construire des tournées de véhicules a fait l'objet de travaux dans la littérature, la prochaine section est consacrée à ce sujet.

2.4 Tournées de véhicules stochastiques/dynamiques

2.4.1 Classification

Les problèmes de tournées de véhicules suscite l'intérêt de très nombreux chercheurs de depuis des décennies. Comme évoqué dans la section précédente, un nombre incalculable de déclinaisons ont été étudiées, combinant différentes contraintes, et visant différents objectifs.

Les VRP, qui sont omniprésents dans la vie de tous les jours, ont d'abord été étudiés dans un contexte statique. Cela signifie que dans les approches proposées, il était fait l'hypothèse que les données sur lesquelles allaient s'appuyer les algorithmes pour construire des tournées étaient statiques, connues avec certitude, et qu'il n'y avait pas de risque qu'elles évoluent avec le temps. Cependant, en se penchant sur des problèmes concrets, on se rend compte que cette hypothèse est rarement respectée dans des situations de la "vie réelle", où des événements imprévus se produisent très souvent en cours de route. Dans des domaines comme l'industrie, le transport ou la santé, un besoin de flexibilité et de dynamisme dans

la manière de planifier les tournées s'est fait ressentir, d'où l'intérêt croissant porté par la recherche aux problèmes de tournées de véhicules impliquant des données non statiques. [Pillac et al., 2013] proposent une classification des VRP selon la nature des données considérées. Pour un VRP donné, ils distinguent ainsi deux critères pour qualifier la nature des données à notre disposition :

La qualité de l'information : Les informations (les données) qu'on utilise pour planifier des tournées peuvent être :

déterministes. Cela signifie qu'elles sont connues de façon certaine (exemple : je sais avec certitude que j'aurai 10 clients à visiter durant ma tournée), ou,

stochastiques. Cela veut dire que les informations en notre possession ne sont connues que de façon partielle, sous forme de probabilités par exemple (j'ai 90% de chances de devoir visiter 10 clients, mais 10% de chances de devoir traiter un client de plus...).

L'évolution de l'information : Les données utilisées pour construire les tournées sont :

- uniquement connues avant le début des tournées, ou,
- également en cours de route, au fur et à mesure des visites.

Ainsi, en combinant ces deux critères, un VRP peut être :

statique et déterministe : Ici, toutes les données sont connues à l'avance, et avec certitude. Aucun évènement imprévu en cours de route n'est à envisager, la tournée planifiée a priori et donc celle qui est effectivement effectuée par le(s) véhicule(s). Il s'agit du type de problèmes le plus étudié dans la littérature, mais également celui qui correspond le moins à des cas réels ;

statique et stochastique : L'information est connue partiellement, et uniquement avant le début des tournées. En d'autres termes, pour construire une tournée, il est possible de s'appuyer sur les données stochastiques que l'on possède à l'avance (des données probabilistes par exemple) pour anticiper d'éventuels imprévus, mais il n'est pas possible d'avoir accès à de nouvelles informations pendant les tournées (on parle de tournées *offline*, car la notion de temps réel n'existe pas) ;

dynamique et déterministe : Les données considérées sont connues avec certitude, et de nouvelles données peuvent être révélées en temps réel pendant le parcours.

Il est donc possible d'exploiter les données qui arrivent au fil de l'eau pour mettre à jour une tournée, s'adapter à de nouveaux événements... Ce type de problèmes a gagné en intérêt auprès de la communauté scientifique avec la démocratisation des technologies de communication et de géolocalisation en temps réel (internet mobile, GPS...) qui sont nécessaires aux traitements des nouvelles informations qui arrivent durant le parcours du véhicule (on parle de tournées *online*) ;

dynamique et stochastique : Ce cas est similaire au précédent, à la différence près que certaines données peuvent être partielles à un moment, avant d'être révélées complètement en cours de route.

2.4.2 Sources de dynamisme

Pour étudier n'importe quel VRP dans un contexte dynamique, il est nécessaire d'identifier, parmi les données nécessaires à la construction des tournées, celles qui peuvent être dynamiques justement. Ici, ce qu'il faut entendre par données dynamiques, ce sont les données qui sont incomplètes, imprécises, ou absentes avant, et qui ne sont révélées avec certitude que pendant les tournées. Ainsi, dans la littérature, quatre principales variables liées aux VRP sont souvent considérées comme dynamiques :

a) Les nœuds à visiter

Dans beaucoup d'applications réelles, de nouvelles requêtes de clients peuvent arriver à n'importe quel moment pendant la tournée. Il est donc impossible de prévoir à l'avance combien de nœuds devront être visités au total, et encore moins où se trouvent ces nœuds. Parmi les travaux considérant l'arrivée dynamique de nouveaux clients, on peut citer [Attanasio et al., 2004] ou encore [Larsen et al., 2004]. Les premiers comparent plusieurs variantes de recherches Tabou pour résoudre un problème où de nouveaux clients demandent, au fur et à mesure, à être transportés d'un point A à un point B (il s'agit d'un *Dial a Ride Problem*, problème décrit dans la section précédente), l'objectif étant de traiter un maximum de demandes tout en minimisant les coûts de service. [Larsen et al., 2004] s'intéressent pour leur part à minimiser le temps d'attente entre l'arrivée d'une demande de client imprévue, et le traitement de la demande.

b) Les durées de trajet

Lorsqu'il s'agit de se déplacer d'un point A à un point B, la durée du trajet correspondante constitue, intuitivement, la première source d'incertitude. Cette variable dépend en effet souvent de facteurs imprévisibles et incontrôlables (route déviée obligeant le véhicule à emprunter un chemin plus long, accident qui bloque la route...). Beaucoup de travaux se sont donc penchés sur les VRP en prenant en compte la caractéristique incertaine des durées de trajet à effectuer entre les points visités. À titre d'exemple, citons [Haghani and Jung, 2005] qui considèrent non seulement des durées de trajet dynamiques, mais également des nouveaux clients qui arrivent dynamiquement, le tout appliqué à une flotte de véhicules.

c) Les demandes des clients

Ici, c'est la "quantité" de biens devant être livrée à chaque nœud qui n'est pas connue de façon certaine à l'avance. Étant donné que le véhicule transporte une quantité limitée de ces biens, s'il venait à en manquer parce qu'un client en a pris une plus grande quantité que prévu, il serait dans l'obligation de ne pas satisfaire les prochains clients de la tournée, ou de se réapprovisionner pour pouvoir poursuivre ses livraisons. Dans tous les cas, le choix de visiter tel nœud avant tel autre peut avoir un impact sur le nombre de demandes traitées, et/ou sur la distance totale parcourue par le véhicule (la coût du détour effectué pour se réapprovisionner diffère selon l'endroit où le véhicule se trouve lorsqu'il doit l'effectuer). Parmi les travaux se penchant sur des demandes de clients inconnues à l'avance, on trouve [Secomandi, 2000] ou [Novoa and Storer, 2009] entre autres.

d) La disponibilité des véhicules

La variable incertaine ici est l'état des véhicules considérés dans la tournée. Il s'agit par exemple de prendre en compte les risques de survenue de pannes qui pourraient mener à une immobilisation temporaire ou définitive des véhicules. Dans les travaux de [Li et al., 2009b] et [Li et al., 2009a], il est par exemple question d'un problème dans lequel des véhicules se voient chacun assigner des nœuds à visiter, et où un des véhicules tombe en panne. Il s'agit alors de réassigner les nœuds associés au véhicule en panne à ceux qui ne le sont pas, et ce de la façon la plus efficace possible. La fonction objectif considérée prend en compte trois critères : les coûts de service, le nombre de visites annulées, et le niveau de perturbation induit par le véhicule en panne sur les parcours respectifs des

autres véhicules.

2.4.3 Approches de résolution

Comme évoqué plus haut, la classification de [Pillac et al., 2013] permet de distinguer quatre types de VRP : les statiques déterministes, les statiques stochastiques, les dynamiques déterministes, et les dynamiques stochastiques. Nous nous intéressons ici aux trois derniers types.

a) Exploiter les données arrivant en temps réel

Pour les problèmes dynamiques déterministes, on ne manipule pas de données stochastiques. En d'autres termes, les informations que l'on possède avant le début d'une tournée sont connues avec certitude, ou bien révélées au fil de l'eau pendant la tournée. Ici, il n'y a donc pas de variables aléatoires que l'on pourrait estimer ou prédire à l'aide de données probabilistes par exemple. Cependant, on a accès, en temps réel, aux nouvelles données qui étaient inconnues au départ, et qui sont révélées en cours de route. On peut donc exploiter ces informations qui arrivent pour s'adapter à de nouveaux contextes, et mettre à jour les tournées en conséquence.

Des approches ont été étudiées pour tirer profit des données récupérées au fur et à mesure du déroulement d'une tournée. Une des plus étudiées est connue sous le nom de "réoptimisation périodique" (*periodic reoptimization*). Elle consiste à planifier des tournées à priori (avant le départ des véhicules), puis à lancer ponctuellement un algorithme d'optimisation des tournées en cours. Celui-ci va alors prendre les données du problème, en prenant en compte celles révélées en cours de route, et va résoudre un problème statique à partir de ces données. Le principal avantage de cette approche est de pouvoir utiliser des algorithmes de résolution de VRP statiques, qui ont été largement étudiés dans la littérature. Il est noter que cette "réoptimisation" des tournées se fait soit de manière périodique (à intervalles réguliers par exemple), ou à l'arrivée de nouvelles informations non prises en compte dans une précédente optimisation. [Psaraftis, 1980] ont proposé un des tous premiers papiers sur ce sujet en étudiant un *Dial a Ride* avec arrivées progressives de nouvelles demandes de clients. Ils ont adopté une approche de résolution exacte, basée sur la programmation dynamique. L'algorithme résolvait périodiquement une instance statique représentant le problème dynamique à un instant t .

L'inconvénient en procédant de cette manière est bien entendu le temps de calcul très important qu'il faut à chaque exécution de l'algorithme, et qui rend une telle approche inadaptée à des problèmes réels qui nécessitent souvent des prises de décision rapides. D'ailleurs, bien qu'il existe dans la littérature d'autres travaux se penchant sur des méthodes de résolution exactes pour des VRP dynamiques, ces travaux restent relativement minoritaires par rapport à ceux ayant recours à des méthodes approchées, et dont les papiers de [Gambardella et al., 2003] et [Montemanni et al., 2005] sont des exemples. Ces derniers proposent un algorithme basé sur des colonies de fourmis pour un VRP dynamique où ils considèrent l'arrivée de nouvelles demandes de clients comme variable incertaine. Leur algorithme d'optimisation est lancé à intervalles de temps réguliers. De cette manière, les nouvelles demandes qui arrivent ne sont prises en compte qu'à la prochaine exécution de l'algorithme (ce qui, par ailleurs, peut être un inconvénient si les nouvelles demandes doivent être traitées urgemment), qui va alors résoudre un problème statique pour construire les prochaines étapes de la tournée.

b) Exploiter les données stochastiques

Lorsque une partie des données du problème à traiter n'est connue que partiellement, à travers une distribution de probabilité par exemple, on parle de problème stochastique. Dans ce cas-là, le problème peut être dynamique stochastique s'il est possible d'exploiter de nouvelles informations qui arrivent en cours de tournées pour revoir ou mettre à jour le tracé de la tournée en cours. Il est statique stochastique s'il n'y a pas moyen de récupérer l'état des données à un instant t pendant la tournée.

Dans tous les cas, des travaux de recherche se sont intéressés à la manière d'exploiter ces données stochastiques pour planifier les tournées, et éventuellement pour les améliorer en temps réel pendant leur déroulement (cas dynamique).

Pour ce faire, l'échantillonnage est l'approche la plus répandue dans la littérature. Ainsi, si l'on considère un problème où des demandes de nouveaux clients peuvent survenir de façon imprévue en cours de route, et qu'on possède, par exemple, la probabilité de survenue de ces nouvelles demandes (en d'autres termes, si on considère l'arrivée de nouvelles demandes comme une variable aléatoire X dont on connaît la distribution), alors, l'échantillonnage consiste à simuler plusieurs scénarios de tournées possibles, impliquant à chaque fois, les demandes de clients déjà formulées (les demandes faites avant la construction de la

ournée), ainsi que des demandes hypothétiques qui pourraient survenir en cours de route, et qui sont des réalisations de la variable aléatoire X .

La multiplication du nombre de scénarios générés permet de couvrir un large éventail de situations pouvant se produire après le début du parcours. La tournée initiale que devra effectuer le véhicule lorsqu'il quitte le dépôt ne comportera bien entendu que des visites chez les clients ayant déjà effectué des demandes, mais l'itinéraire construit grâce aux différents scénarios étudiés donne la possibilité "d'insérer" dans le parcours, et à moindre coût, d'éventuelles nouvelles visites au fur et à mesure qu'elles arrivent. Il est à noter que la tournée initiale construite n'est pas forcément la plus optimale si on ne considère que les demandes planifiées à l'avance, mais c'est celle qui offre le plus de flexibilité, et qui est la moins impactée en cas d'arrivée de nouvelles demandes imprévues.

Dans le cas où les nouvelles données qui arrivent en temps réel ne sont pas exploitables par l'algorithme (cas statique stochastique), le parcours initial généré (avant le début de la tournée donc) ne peut être amélioré à la lumière du déroulement effectif de la tournée en cours.

En effet, le véhicule, lorsqu'il reçoit une nouvelle demande, ne peut que, éventuellement, se détourner temporairement de son parcours initial afin de satisfaire la nouvelle demande, avant de reprendre son chemin planifié à l'avance. Aucune "réoptimisation" du parcours initial n'est possible dans ce cas-là. À l'inverse, si les données arrivant en temps réel sont exploitables par l'algorithme (cas dynamique stochastique), alors celui-ci peut, lorsqu'une "vraie" nouvelle demande arrive, mettre à jour les scénarios qu'il avait générés au départ, en créer de nouveaux prenant en compte cette nouvelle demande, supprimer des scénarios obsolètes (incompatibles avec l'état actuel des choses) pour, ainsi, mettre à jour le parcours du véhicule, qui de cette manière, n'est plus seulement amené à faire des détours ponctuels de son itinéraire initial, mais également à voir la suite de son parcours modifiée pour tenir compte des nouvelles informations révélées.

Parmi les travaux adoptant cette approche par échantillonnage, on peut citer [Chang et al., 2000, Bent and Van Hentenryck, 2004c, Bent and Van Hentenryck, 2004a, Bent and Van Hentenryck, 2004b, Flatberg et al., 2007].

2.4.4 Synthèse

Les VRP ont été largement étudiés dans la littérature depuis des décennies. De nouvelles combinaisons de contraintes et d'objectifs sont introduites en permanence dans les nombreuses études qui se succèdent, et ce dans le but de coller au maximum à la réalité du terrain, et aux besoins concrets du monde de l'industrie, des transports et des communications, entre autres.

Partant de là, la nécessité d'inclure dans les travaux de recherche les notions de dynamisme et d'incertitude qui entourent certains aspects des tournées de véhicules "réelles" s'est vite fait ressentir, et a donné naissance à une multitude de travaux se penchant donc sur les VRP dynamiques.

Les VRP dynamiques sont ceux pour lesquels une partie des données n'est pas connue ou est connue de façon imprécise à l'avance (avant le début des tournées). Ce sont des VRP pour lesquels des évènements imprévus peuvent survenir à n'importe quel moment durant l'exécution d'une tournée. [Pillac et al., 2013] proposent une classification des VRP selon la disponibilité et la qualité des données nécessaires à la construction des tournées (voir figure 2.6 et section 2.4 pour plus de détails).

Les principales variables qui peuvent générer de l'incertitude et du dynamisme dans un VRP sont :

- l'arrivée de nouveaux nœuds à visiter pendant une tournée ;
- les durées de trajet variables entre les nœuds visités ;
- les demandes de clients variables ;
- l'état des véhicules (la survenue de pannes notamment).

S'agissant du schéma logistique étudié dans ce chapitre, l'incertitude est engendrée par l'état des bacs à béton récupérés chez les clients visités. Ainsi, si un bac est récupéré, et que des résidus de béton non consommé s'y trouvent, alors le véhicule se voit dans l'obligation d'acheminer ces résidus de béton en vue de les recycler dans un des centres de recyclage à proximité.

Nous pouvons considérer, pour notre cas, que les centres de recyclage sont des "clients" qui peuvent faire des demandes de visite à n'importe quel moment durant une tournée. Ainsi, à chaque fois qu'un bac est collecté et qu'il n'est pas vide, les centres de recyclage à proximité sont vus comme des nœuds requérant une visite immédiate. Le but est alors

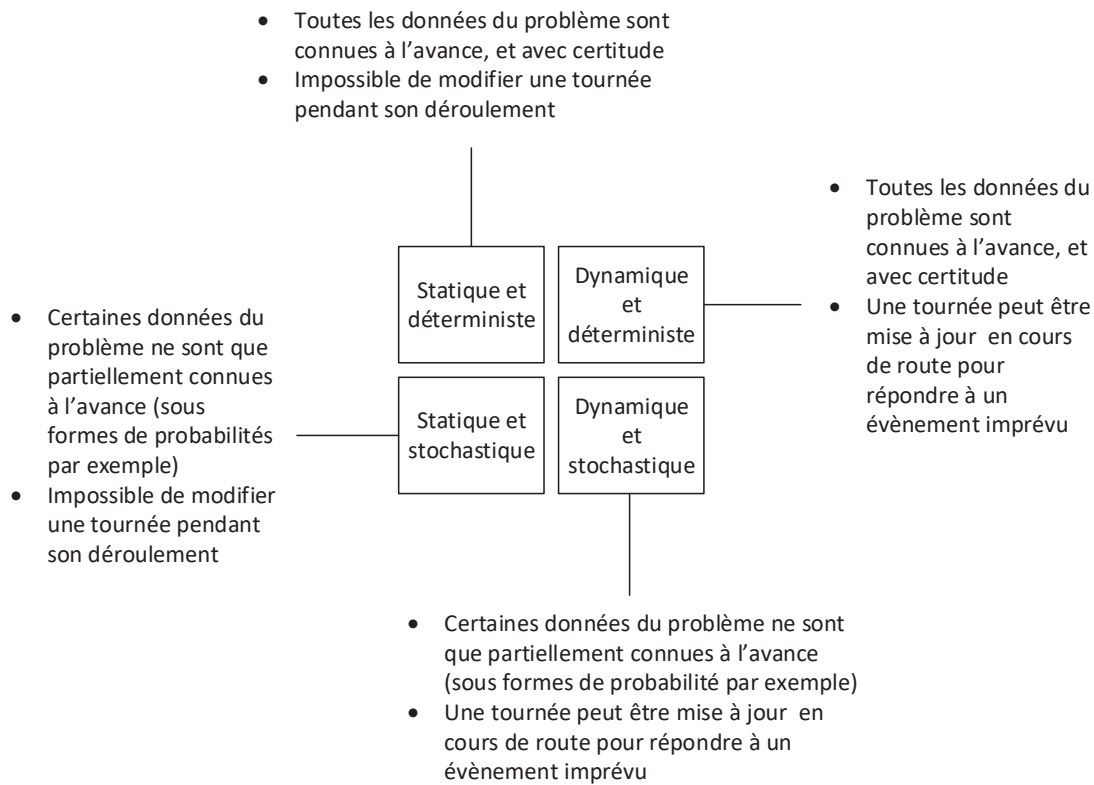


FIGURE 2.6 – Classification des VRP selon la qualité et l'évolution des données [Pillac et al., 2013]

de satisfaire une de ces requêtes, la plus pratique et la moins pénalisante pour la suite du parcours du véhicule, et de rejeter les autres.

Pour rappel, les tournées induites par le schéma logistique étudié ici sont des tournées avec collectes et livraisons faisant appel à un unique véhicule, et qui définissent une structure 1-M-1 (voir section 2.3) où les nœuds visités requièrent une livraison ou une collecte mais pas les deux.

À notre connaissance, il n'y a pas de travaux abordant ce type de tournées dans un contexte dynamique. Aussi, nous proposons deux approches pour répondre à notre problème. Dans la première, l'approche statique stochastique, nous exploitons des probabilités de passages en centres de recyclage pour essayer, à priori, d'anticiper les futurs détours qui pourraient survenir, et réduire ainsi leur impact sur la qualité des tournées, et cela via une approche par échantillonnage (voir section 2.4.3). La seconde approche quant à elle, la dynamique déterministe, consiste à adapter les tournées en cours de route pour "réoptimiser" à chaque fois qu'un passage en centre de recyclage est nécessaire. Il s'agit donc d'une approche de réoptimisation périodique (voir 2.4.3 pour plus de détails).

Les deux approches sont détaillées dans la suite de ce chapitre.

2.5 Formalisation du problème

La problème tel que présenté dans la section 2.1 est un problème 1-M-1 à un véhicule, et avec présence de données inconnues à l'avance, liées à la question des détours imprévus en centres de recyclage.

Étant donné un véhicule, dont la capacité maximale de chargement Q , exprimée en nombre de bacs, est finie et connue, et étant donné :

- N_d , l'ensemble des points où des bacs doivent être livrés, avec $Card(N_d) = n_d$;
- N_p , l'ensemble des points où des bacs doivent être collectés, avec $Card(N_p) = n_p$;
- N_r , l'ensemble des centres de recyclage vers lesquels peuvent être, potentiellement, acheminés les résidus de béton non consommé récupérés dans les bacs collectés. $Card(N_r) = n_r$;
- O , le point de départ et d'arrivée du véhicule effectuant la tournée (le dépôt).

Avec :

- $N_d \cap N_p = \emptyset$;
- $N_r \cap N_p = \emptyset$;
- $N_r \cap N_d = \emptyset$;
- $O \notin N_d \cup N_p \cup N_r$.

Le problème peut être défini sur un graphe complet $G = (V, E)$ où :

- $V = N_d \cup N_p \cup N_r \cup \{O\}$. En d'autres termes, l'ensemble des sommets du graphe G représente tous les points **devant être visités** ($N_d \cup N_p \cup \{O\}$), ainsi que tous les points **pouvant être visités** (N_r) durant une tournée. $Card(V) = n_p + n_d + n_r + 1$ représente le nombre de sommets de G .
- $E = \{(i, j), i, j \in V, i \neq j\}$. L'ensemble des arêtes de G représente les chemins reliant les différents points considérés dans la tournée.

En considérant les ensembles suivants, associés au graphe G :

- $C = \{c_{i,j}, (i, j) \in E\}$ où $c_{i,j}$ est la distance entre i et j (avec $c_{i,j} = c_{j,i}$) ;
- $D = \{d_i, i \in V\}$ où $|d_i|$ est représentée la demande en nombre de bacs du point i ($d_i > 0$ pour les livraisons de bacs et $d_i < 0$ pour les collectes) ;

Et en définissant :

- q_i , le nombre de bacs dans le véhicule après son passage par le point i ;
- $x_{i,j}$, une variable booléenne telle que :
 - $x_{i,j} = 1$ si et seulement si le véhicule qui assure la tournée visite le point j immédiatement après avoir visité le point i ;
 - $x_{i,j} = 0$ sinon.

Notre objectif est de minimiser le coût global de la tournée effectuée par le véhicule :

$$\min \sum_{i=0}^n \sum_{j=0}^n x_{i,j} c_{i,j} \quad (2.1)$$

tout en respectant un certain nombre de contraintes :

$$\sum_{j \in V} x_{i,j} = 1 \quad \forall i \in N_d \cup N_p, i \neq j \quad (2.2)$$

$$\sum_{i \in V} x_{i,j} = 1 \quad \forall j \in N_d \cup N_p, i \neq j \quad (2.3)$$

$$x_{i,j} = 0 \quad \forall i \in N_r \cup N_d \cup O, \forall j \in N_r \quad (2.4)$$

$$q_i - x_{i,j} d_j \leq Q \quad \forall (i, j) \in V \quad (2.5)$$

$$q_i \geq 0 \quad \forall (i, j) \in V \quad (2.6)$$

$$q_i - d_j \leq M(1 - x_{i,j}) + q_j \quad \forall (i, j) \in V \quad (2.7)$$

$$q_i - d_j \geq -M(1 - x_{i,j}) + q_j \quad \forall (i, j) \in V \quad (2.8)$$

$$q_o = Q_0 \quad (2.9)$$

$$q_i \in \mathbb{N} \quad \forall (i, j) \in V \quad (2.10)$$

$$x_{i,j} \in \{0, 1\} \quad \forall (i, j) \in V \quad (2.11)$$

$$M \in \mathbb{N} \quad (2.12)$$

- Les contraintes 2.2 et 2.3 garantissent que chaque point de passage obligatoire (point de livraison ou point de collecte) ne peut être visité qu'une et une seule fois ;
- La contrainte 2.4 indique que le véhicule ne peut se diriger vers un centre de recyclage que s'il vient de récupérer des bacs contenant du béton non utilisé dans un des points de collecte prévus dans la tournée. Sont donc interdits, les trajets menant vers les centres de recyclage depuis :
 - le point de départ de la tournée (le dépôt) ;

- un point de livraison ;
- ou un autre centre de recyclage ;
- La contrainte 2.5 est liée à la capacité maximale du véhicule.

Ainsi :

- si un point j est visité immédiatement après un point i , alors :

$$x_{i,j} = 1 \implies q_i - d_j \leq Q$$

En d'autres termes, dans le cas où le point j est un point de collecte, il ne faudrait pas que le nombre de bacs récupérés, représenté par d_j , additionné au nombre de bacs présents dans le véhicule (q_i), dépasse la capacité maximale Q du véhicule. Pour rappel, par convention, $d_j < 0$ dans le cas d'une collecte, et > 0 s'il s'agit d'une livraison ;

- si j n'est pas visité immédiatement après i , alors :

$$x_{i,j} = 0 \implies q_i \leq Q$$

Ce qui signifie qu'à tout moment, le nombre de bacs chargés dans le véhicule ne doit pas dépasser la limite Q . De ce fait, un véhicule ne peut rejoindre un point de collecte si le nombre de bacs à y récupérer, ajouté au nombre de bacs déjà dans le véhicule, dépasse la capacité maximale autorisée. Dans ce cas-là, le véhicule ne peut que se diriger vers des points de livraison pour libérer de l'espace pour les prochaines collectes.

- La contrainte 2.6 indique que le nombre de bacs à l'intérieur du véhicule ne peut être négatif, ce qui serait absurde ;
- Les contraintes 2.7 et 2.8 définissent la variable q_i , c'est-à-dire le nombre de bacs présents à l'intérieur du véhicule après avoir quitté le point i . Ainsi :
- Si un point j est visité dans la foulée d'un point i , alors :

$$x_{i,j} = 1 \implies (q_i - d_j \leq q_j \wedge q_i - d_j \geq q_j) \implies q_j = q_i - d_j$$

Autrement dit, lorsque j est visité dans la foulée de i , alors le nombre de bacs restant dans le véhicule lorsqu'il quitte le point j (noté q_j), est égal au nombre

de bacs chargés dans le véhicule lorsqu'il quitte i (q_i), moins le demande de j (d_j);

- Si j n'est pas visité immédiatement après i , alors :

$$x_{i,j} = 0 \implies -M \leq q_i - d_j - q_j \leq M$$

Ce qui est toujours vérifié car M (resp. $-M$) est un entier positif (resp. négatif) suffisamment grand (resp. petit) pour que l'équation soit toujours vraie (on peut prendre, $M = Q + \sum |d_j|$)

- La contrainte 2.9 affecte au véhicule une charge initiale de Q_0 bacs à sa sortie du dépôt,

Le schéma 2.7 illustre les différentes variables associées aux points de passage d'une tournée.

2.6 Contribution

Après avoir décrit le schéma logistique à traiter (section 2.1), et formalisé les objectifs et contraintes qui l'entourent (section 2.5), nous exposons ci-dessous les approches que nous avons adoptées pour mettre en œuvre un système capable de générer des tournées conformes aux exigences du schéma présenté, qui respectent les contraintes détaillées plus haut, prennent en compte l'impact d'éventuels détours imprévus en cours de route, et minimisent, autant que faire se peut, les coûts de trajet associés.

Les problèmes de tournées de véhicules sont généralement difficiles à résoudre de façon exacte et en un temps raisonnable. Dans notre cas, cela est encore plus vrai que pour un problème classique dans le sens où nous devons traiter des livraisons et des collectes, le tout avec un véhicule ayant une capacité de transport limitée. C'est pourquoi, nous avons préféré les méthodes approchées aux méthodes exactes pour aborder notre problème. Les méthodes approchées, représentées notamment par les métaheuristiques, présentent en effet un compromis intéressant entre qualité des solutions obtenues et temps d'exécution. Nous proposons ainsi deux algorithmes basés sur le principe de recherche locale. Le premier exploite des données stochastiques pour anticiper les éventuels détours qui pourraient survenir durant la future tournée et construire, en conséquence, un itinéraire qui minimise l'impact de ces détours sur le coût global de la tournée. Cet algorithme planifie des

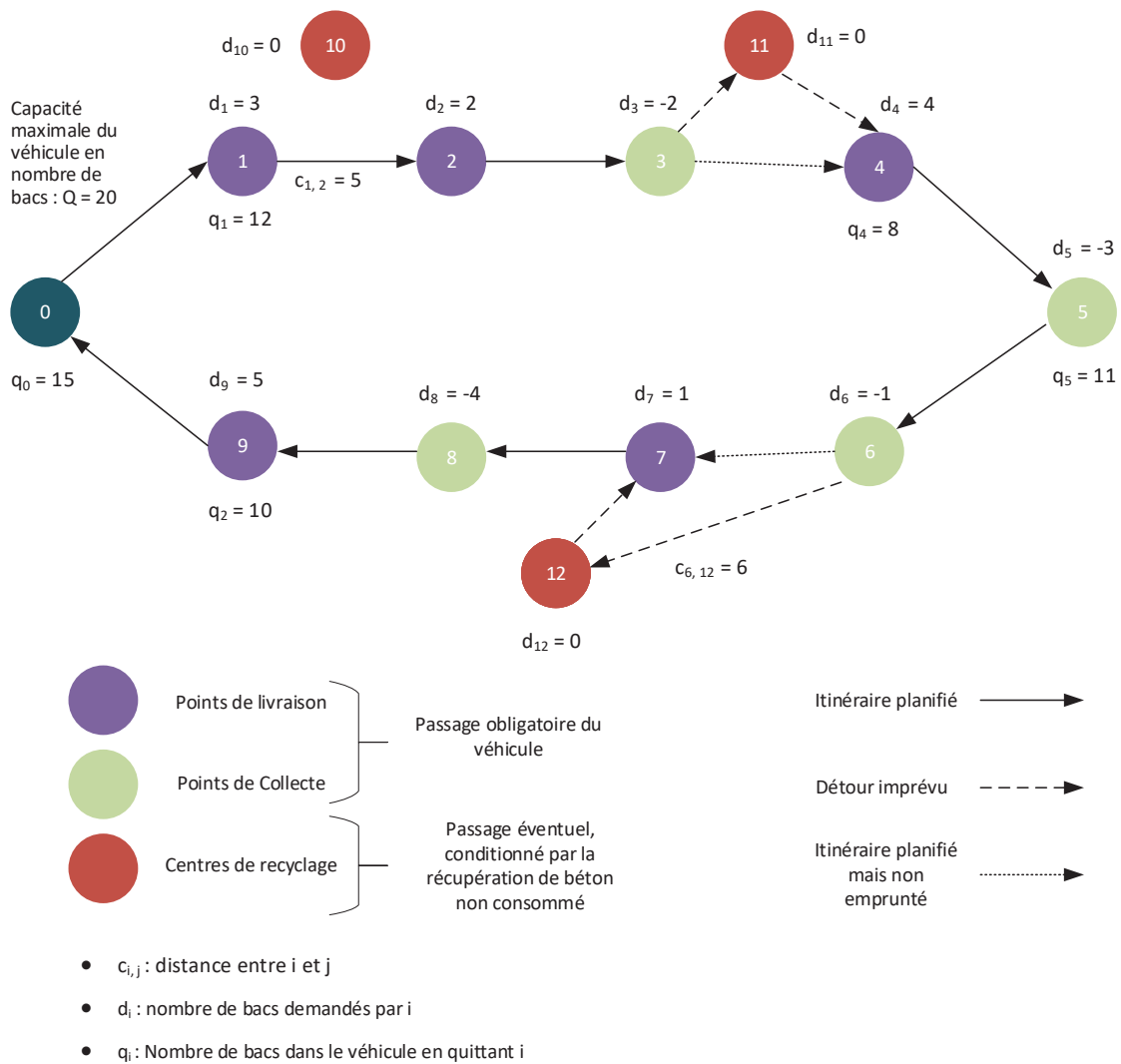


FIGURE 2.7 – Graphe représentant une tournée de livraison et de collecte de bacs avec détours en centres de recyclage

tournées "à priori". Autrement dit, il génère l'itinéraire d'une tournée avant que le véhicule associé ne quitte le dépôt. Les tournées ainsi créées ne sont pas amenées à changer en cours de route, l'ordre dans lequel les différents points de passage sont visités est statique, et seuls des passages en centre de recyclage peuvent, éventuellement, s'intercaler entre les points de passage obligatoires. On parle alors de tournées statiques et stochastiques. Statiques car l'ordre préétabli des points visités ne change pas, stochastique en raison du fait qu'une partie des données utilisées pour générer les tournées n'est connue que partiellement, par le biais de probabilités.

La seconde approche proposée est plus dynamique en ce sens que l'ordre dans lequel les points de passage doivent être visités peut changer en cours de route en fonction des détours qui peuvent rendre le chemin préalablement choisi plus coûteux qu'un autre,

calculé pendant la tournée. On parle ici de tournées dynamiques car l'itinéraire préétabli est amené à évoluer au fur et à mesure de l'obtention d'informations sur le nombre de détours à effectuer, et les lieux où ils se produisent.

2.7 Approche stochastique

Les passages en centres de recyclage que peut être amené à effectuer le véhicule pendant sa tournée sont nécessaires :

- d'une part, pour éviter aux clients d'avoir à gérer eux-mêmes les résidus de béton qu'ils n'ont pas consommés ;
- et d'autre part, pour éviter le gaspillage du produit non consommé en donnant la possibilité aux producteurs de béton de le recycler pour d'autres usages.

Néanmoins, le temps que passe le véhicule à se détourner de son itinéraire principal pour acheminer des bacs en centres de recyclage, est autant de temps qu'il n'exploite pas à servir les clients. Par conséquent, et pour garantir une tournée efficace et rentable, il est crucial de minimiser l'impact des coûts de trajet liés aux détours sur le coût global de la tournée.

Pour ce faire, nous proposons ici une approche, adaptée du travail de [Birattari et al., 2008] sur une variante du problème du voyageur de commerce dans laquelle certains points à visiter peuvent, durant la tournée, et avec une certaine probabilité, ne plus requérir de visites du véhicule.

2.7.1 Principe

L'algorithme se base sur une recherche locale qui part d'une tournée initiale (on parlera de solution initiale) S , et tente de l'améliorer, itération après itération, en cherchant des solutions "voisines" de meilleure qualité. Afin d'éviter les optimum locaux, la recherche locale est lancée plusieurs fois, avec différentes solutions initiales.

Algorithme 1 : Principe de la recherche locale

```

bestSolution ← null ;
for i in range(1, NbInitialSolutions) do
    improved ← true ;
    /* Affecter la solution initiale */
    Si ← initialSolution(i) ;
    /* Tant que la solution courante est améliorée... */
    while improved do
        improved ← false ;
        /* Générer le voisinage de Si */
        N ← Neighborhood(Si) ;
        /* Éliminer les solutions incohérentes du voisinage N */
        N ← removeUnfeasibleSolutionsNeighborhood(N) ;
        for S' in N do
            /* Remplacer la solution courante par une solution voisine si la
               valeur de la fonction objectif de cette dernière est inférieure */
            if  $f(S') < f(S_i)$  then
                Si ← S' ;
                improved ← true ;
        /* Choisir la meilleure solution parmi toutes celles obtenues avec les
           différentes solutions initiales */
        if bestSolution = null ||  $f(S_i) < f(\textit{bestSolution})$  then
            bestSolution ← Si ;

```

Pour détailler le fonctionnement de cette approche, il est nécessaire de répondre à un certain nombre de questions que l'on peut se poser en parcourant l'algorithme 1 :

1. Qu'est-ce qu'une "solution" ;
2. Comment sont générées les solutions initiales utilisées comme points de départ de la recherche ?
3. Comment est défini le voisinage d'une solution ?
4. Qu'est ce qu'une solution incohérente ?
5. Comment calcule-t-on $f(S)$, la valeur de la fonction objectif pour la solution S ?

2.7.2 Définition de la solution

La solution est une représentation algorithmique d'une tournée de véhicule. Il s'agit simplement d'un vecteur dont chaque élément est un des points à visiter durant la tournée. Le vecteur est ordonné de sorte que le $i^{\text{ème}}$ élément représente le point visité à la $i^{\text{ème}}$ position par le véhicule (voir figure 2.8).

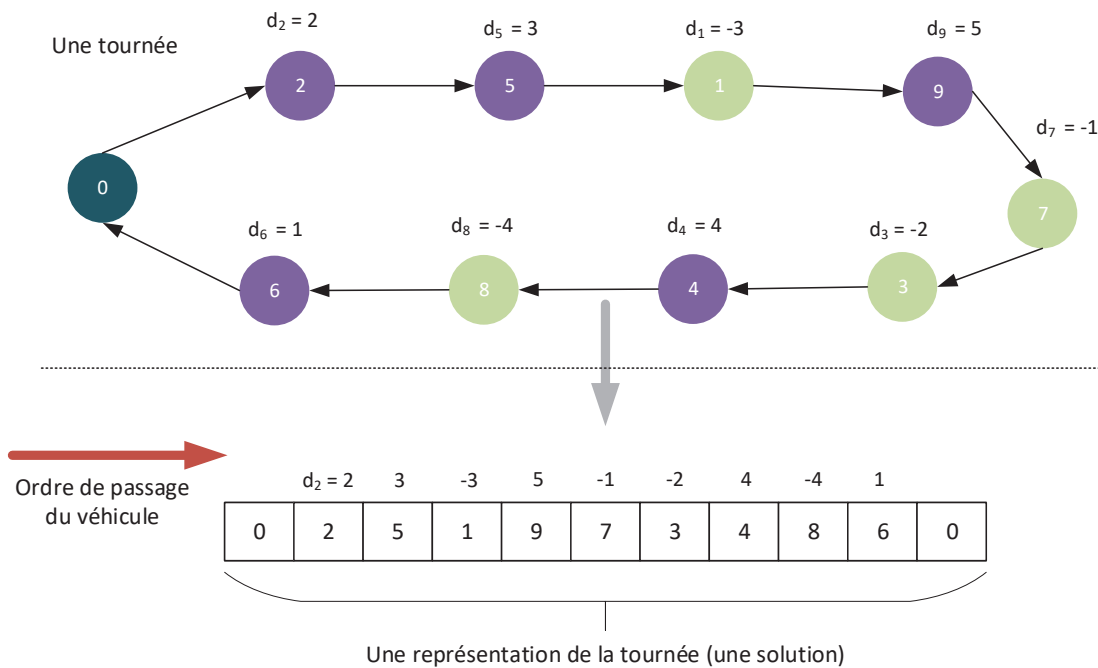


FIGURE 2.8 – Représentation d'une tournée

2.7.3 Choix des solutions initiales

Une solution initiale est une tournée cohérente construite pour être la première dont le voisinage est exploré. Le choix de la solution initiale conditionne donc le chemin effectué par l'algorithme dans sa recherche de la meilleure solution possible.

Remarque 2. *Il est à notre qu'à ce stade, on ne se soucie pas encore des détours imprévus. Il s'agit uniquement de générer des tournées cohérentes, qui prennent en compte toutes les contraintes énoncées dans le cas idéal (cas où il n'y a aucun imprévu en route).*

En explorant le voisinage d'une solution, le risque est que cette dernière soit un optimum local, et que la recherche dans son voisinage ne permette pas de trouver une solution

meilleure, dont on pourrait explorer à nouveau le voisinage à l'itération suivante pour essayer de l'améliorer... Pour éviter à l'algorithme de se retrouver piégé dans un optimum local, nous lui soumettons plusieurs solutions initiales, la recherche locale étant effectuée à partir de chacune des solutions fournies. Le but ici est, d'abord, d'élargir l'espace de recherche en démarrant de points différents, puis, d'explorer plus finement chacun des sous-espaces autour des points de départ choisis grâce au mécanisme de recherche locale. Ainsi, plusieurs solutions initiales sont générées, certaines aléatoirement, en veillant à ce que tous les points de livraison et de collecte soient visités, et qu'à aucun moment, la contrainte de capacité liée au véhicule ne soit violée. D'autres solutions sont d'autre part générées grâce à des heuristiques simples, dont voici quelques exemples :

a) Heuristique 1 : Priorité aux points les plus "proches"

Ici, il s'agit de construire une tournée où le véhicule se déplace toujours du point courant i vers le point j qui minimise $c_{i,j}$. En d'autres termes, à chaque fois qu'un point est visité, le point suivant est choisi de sorte à minimiser le coût du trajet à venir (voir algorithme 2).

Une fois la tournée générée, on vérifie que la contrainte de capacité liée au véhicule est

Algorithme 2 : Recherche du plus proche voisin

```

/* La tournée démarre du point  $i = 0$  */
 $i \leftarrow 0$  ;
 $solution \leftarrow [0]$  ;
/* Initialement, l'ensemble des points à visiter est composé de l'union des
points de collecte et des points de livraisons */
 $remainingPoints \leftarrow N_p \cup N_d$  ;
/* Tant que tous les points prévus n'ont pas tous été visités... */
while  $remainingPoints \neq \emptyset$  do
     $minCost \leftarrow 0$  ;
    /* Choisir le point non encore visité qui minimise le coût du prochain
trajet */
    for  $j$  in  $remainingPoints$  do
        if  $minCost = 0$  OR  $minCost > c_{i,j}$  then
             $nextPoint \leftarrow j$  ;
    /* Mettre à jour le point courant  $i$ , la solution, et l'ensemble des points
non encore visités */
     $i \leftarrow nextPoint$  ;
     $solution \leftarrow solution.push(i)$  ;
     $remainingPoints \leftarrow remainingPoints - \{i\}$  ;

```

toujours respectée en cours de route, ce qui n'est pas évident à priori. Si est elle cohérente, la tournée est alors retenue dans la liste des solutions initiales soumises à la recherche

locale, sinon, elle est soumise à une série de permutations visant à éliminer l'incohérence. Concrètement, si une violation de la contrainte de capacité est détectée au niveau de l'élément i du vecteur représentant la tournée, cela signifie qu'en quittant le $i^{\text{ème}}$ point, le véhicule a collecté un nombre de bacs chez i qu'il ne peut, techniquement, pas entreposer dans le véhicule.

Remarque 3. *Il est à noter qu'une violation de la contrainte de capacité ne peut survenir que lors d'une collecte, car c'est à ce moment que le risque de dépasser la capacité maximale du véhicule existe. Lors d'une livraison, le problème ne se pose pas car le véhicule se décharge d'un certain nombre de bacs.*

Pour essayer de "corriger" une solution qui serait incohérente en un point de **collecte** visité à la $i^{\text{ème}}$ position, le principe est de choisir un point de **livraison** se trouvant à la $j^{\text{ème}}$ position dans l'ordre de passage du véhicule, avec $j > i$, et de l'avancer à la position i , en veillant à décaler d'une position tous les points entre i et $j - 1$ (voir figure 2.9). Le but ici est de permettre au véhicule de décharger plus de bacs avant d'arriver au point de collecte problématique, et d'avoir ainsi plus d'espace, et donc moins de risque de dépassement de la capacité maximale prévue pour le véhicule. Le processus de permutation est répété jusqu'à ce que la solution obtenue soit entièrement cohérente, ou qu'il n'y ait plus de permutations possibles.

b) Heuristique 2 : Priorité aux livraisons

Le principe ici est de générer une tournée au cours de laquelle tous les points de livraison sont visités en priorité par le véhicule. Les collectes ne sont donc effectuées qu'une fois le véhicule complètement vidé des bacs qu'il doit livrer. Ainsi :

- à sa sortie du dépôt, le véhicule se dirige vers le point de **livraison** i qui minimise $c_{0,i}$ puis applique la même règle pour tous les points de livraison ;
- puis, à partir du dernier point de livraison visité i , il se dirige vers le point de **collecte** j qui minimise $c_{i,j}$ et continue ainsi jusqu'au dernier point de collecte, avant de retourner au dépôt.

Remarque 4. *Cette heuristique présente l'avantage de ne pas construire de tournées incohérentes. En effet, étant donné que les livraisons sont toutes effectuées en amont, le véhicule entame ses collectes sans aucun bac à livrer à bord, et donc avec un espace de stockage maximal.*

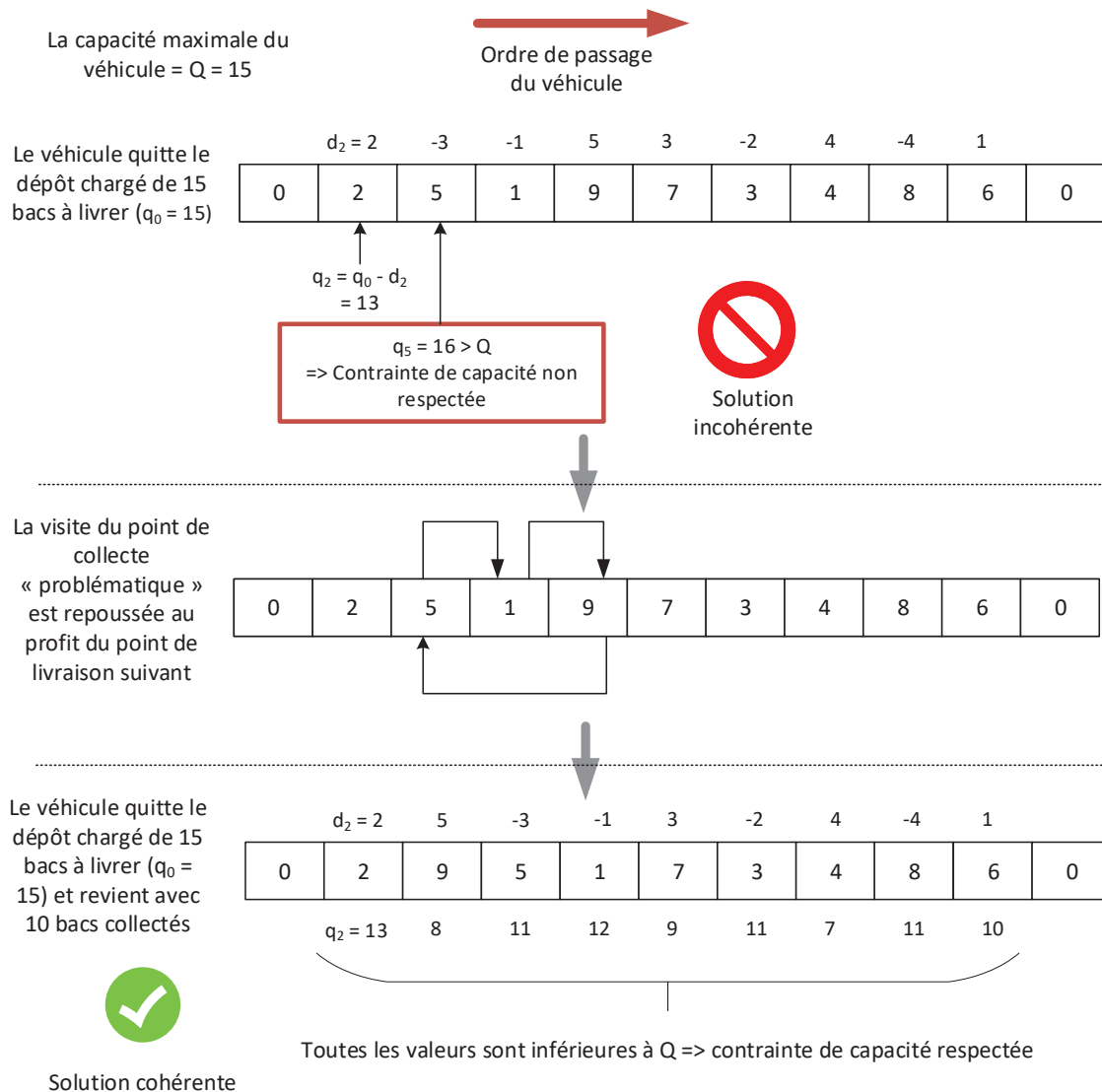


FIGURE 2.9 – Solution incohérente VS Solution cohérente

Remarque 5. Il va sans dire que $Q \geq \sum_{i \in N_p} d_i$. La capacité maximale du véhicule doit toujours être supérieure ou égale au nombre de bacs total à collecter, car au final, le véhicule devra acheminer l'intégralité des bacs ramassés vers le dépôt.

c) Heuristique 3 : Alternance livraison/collecte

Ici, il s'agit d'alterner les opérations de livraison et de collecte de sorte à minimiser les chances d'avoir une solution non conforme à la capacité du véhicule.

Ainsi les points de livraison et les points de collecte sont séparés en deux ensemble distincts. Les points de livraison sont triés suivant l'ordre **décroissant** du nombre de bacs à livrer, et les points de collecte sont triés suivant l'ordre **croissant** du nombre de bacs à

recupérer. Le véhicule passe alors du point de livraison le plus "gourmand" vers le point de collecte le moins "généreux" en bacs, puis vers le deuxième point de livraison et ainsi de suite...(voir figure 2.10).

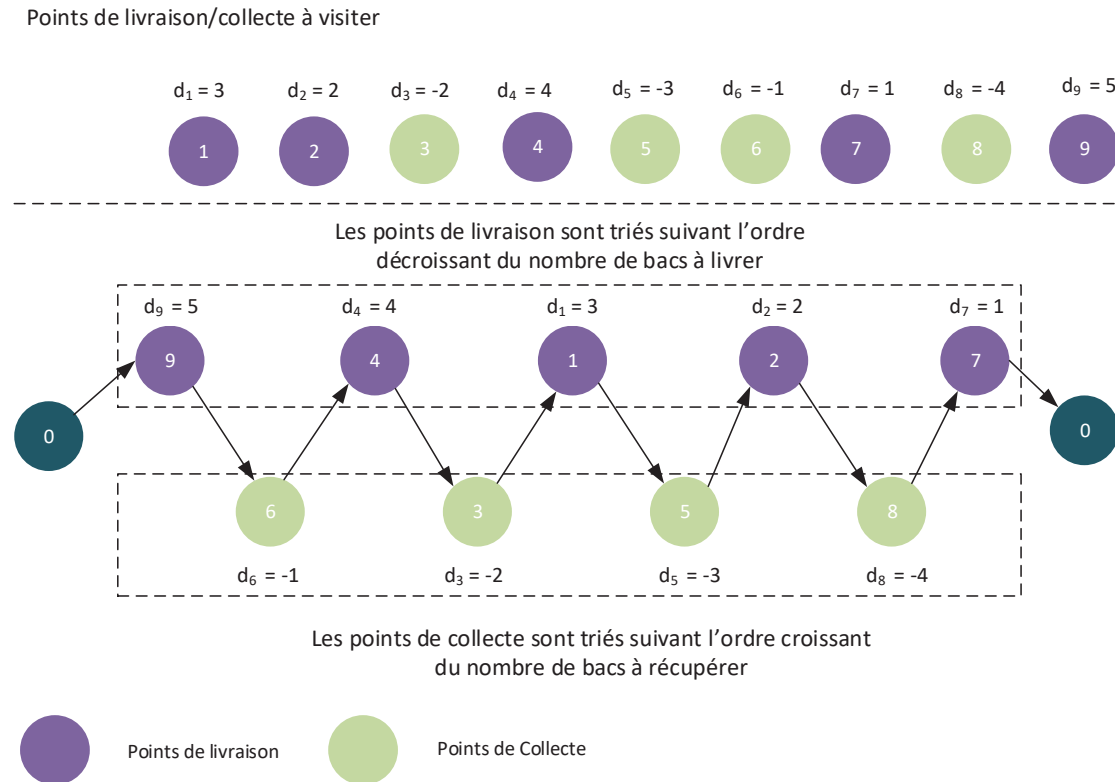


FIGURE 2.10 – Heuristique 3 : Alternance livraison/collecte

De cette manière, théoriquement, le véhicule aurait :

- en début de tournée, plus de bacs à déposer que de bacs à reprendre, ce qui lui permettrait de gagner en espace de stockage et de minimiser le risque de dépassement de sa capacité maximale ;
- puis, en fin de tournée, plus de bacs à récupérer que de bacs à déposer, ce qui devrait moins poser problème grâce à l'espace de stockage gagné en début de tournée.

2.7.4 Définition du voisinage d'une solution

À chaque itération de la recherche locale, l'espace de recherche autour de la solution courante est exploré afin d'essayer de trouver une solution "voisine" de meilleure qualité, qui servira de solution courante lors de l'itération suivante.

Pour ce faire, nous définissons le voisinage V , de taille n , d'une solution S comme un

ensemble de solutions obtenues en soumettant n fois la solution S à l'algorithme *1-Shift*, qui a été proposé par [Bertsimas, 1988], et qui est largement utilisé dans la littérature pour dériver des solutions voisines à partir d'une solution initiale. Cette méthode consiste à choisir, aléatoirement, deux nombres i et j , tels que $i < j$ et que $(i, j) \in [1, n_p + n_d]$, et à déplacer le point se trouvant à la $i^{\text{ème}}$ position vers la $j^{\text{ème}}$. Les points visités entre les positions $i + 1$ et j sont alors avancés d'une position (voir figure 2.11).

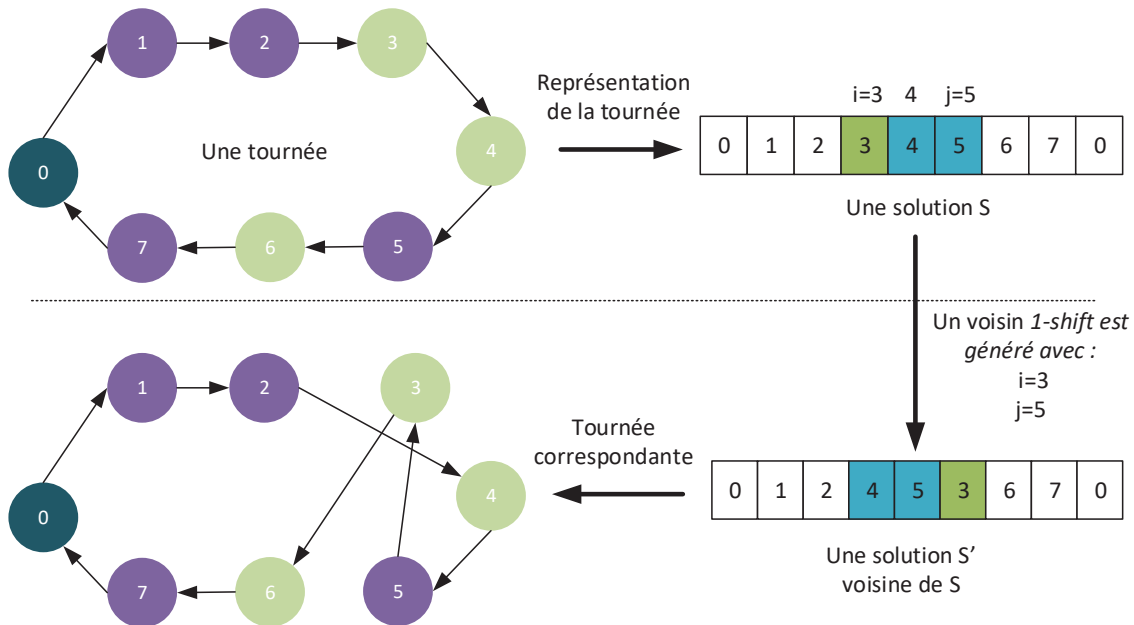


FIGURE 2.11 – Algorithme *1-Shift* : principe

2.7.5 Cohérence d'une solution

Comme évoqué plus haut, une solution est dite cohérente si elle respecte toutes les contraintes de la tournée. Plus précisément, la capacité maximale du véhicule qui ne doit jamais être dépassée pendant une tournée constitue la contrainte la moins évidente à respecter.

En effet, si S' est une solution voisine de S , et si S est une solution cohérente, rien ne garantit, a priori, que S' est cohérente, car rien ne garantit que la contrainte de capacité n'est pas violée dans S' . Pour épurer le voisinage d'une solution, il faut donc procéder à la vérification de chacun de ses éléments, et éliminer les solutions qui ne respectent pas les contraintes énoncées.

Vérifier qu'une solution respecte la contrainte de capacité revient à vérifier que

$$q_i \leq Q, \forall i \in N_p \cup N_d$$

avec :

$$q_i = q_0 - \sum_{j=1}^{j < i} d_j$$

où :

- Q est la capacité de stockage maximale du véhicule (en nombre de bacs), à ne jamais dépasser ;
- q_i est le nombre de bacs restant dans le véhicule lorsqu'il quitte le $i^{\text{ème}}$ point à visiter ;
- d_j est le nombre de bacs à déposer, ou à collecter au $j^{\text{ème}}$ point.

En outre, si S' est un voisin *1-Shift* d'une solution cohérente S , obtenu en décalant un point de la $i^{\text{ème}}$ à la $j^{\text{ème}}$ position, il est facile à démontrer qu'il suffit de vérifier la cohérence de S' entre les points i et $j - 1$ pour garantir qu'elle est entièrement cohérente. En effet, S et S' sont identiques entre les points 0 et $i - 1$. Sachant que S est cohérente sur cette portion (car elle est cohérente globalement), S' est donc également cohérente entre les points 0 et $i - 1$. De la même manière, pour la portion allant de $j + 1$ jusqu'au dernier point visité, S et S' sont identiques. La cohérence de S' est donc conditionnée par la cohérence du trajet entre les points $i + 1, i + 2, \dots, j$ (voir figure 2.12).

2.7.6 Fonction objectif

Rechercher la meilleure solution possible n'a de sens que si l'on définit un critère objectif, permettant de comparer les solutions entre elles. C'est le rôle de la fonction objectif.

La fonction objectif f , appliquée à une solution donnée, retourne une valeur que l'on cherche à minimiser. Ainsi, si S et S' sont deux solutions dont on veut comparer la qualité, on dira que S est **meilleure** que S' si et seulement si $f(S) < f(S')$.

Pour notre problème, la valeur à minimiser est le coût global de la tournée, qui est égal à la somme des coûts engendrés par chaque trajet entre 2 points consécutifs de la tournée.

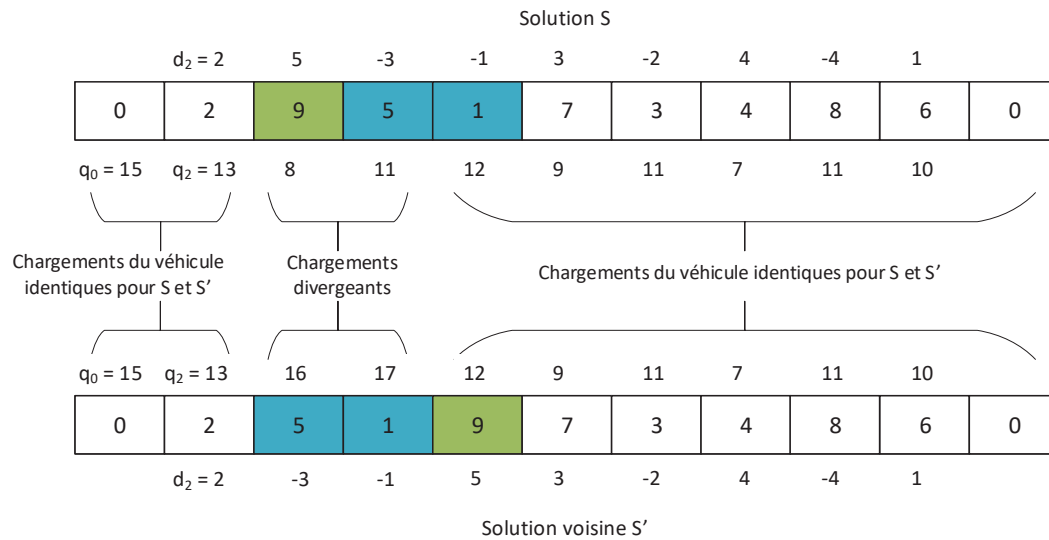


FIGURE 2.12 – Algorithme *1-Shift* : contrainte de capacité

La fonction objectif peut donc s'exprimer de la manière suivante :

$$f(S) = \sum_{i \in N} \sum_{j \in N} x_{i,j} c_{i,j}$$

où :

- N est l'ensemble des points visités ;
- $x_{i,j}$ est un booléen qui est à 1 si j est visité immédiatement après i ;
- $c_{i,j}$ est le coût associé au trajet entre i et j .

Calculer la valeur de la fonction objectif pour une solution donnée peut paraître trivial, mais en pratique, $f(S)$ est une information dont on ne dispose pas avant la fin de la tournée.

En effet, calculer précisément $f(S)$ implique de connaître l'itinéraire exact emprunté par le véhicule durant sa tournée. Or, il se trouve que le véhicule peut être amené à effectuer des détours imprévus en centres de recyclage lorsqu'il collecte des bacs contenant du béton non consommé (voir section 2.1.3).

Les tournées étant soumises à l'incertitude inhérente aux changements d'itinéraire imprévus, la valeur de f devient une inconnue qui n'est révélée qu'à la fin de la tournée. Ainsi, le recours à la fonction f , telle que définie plus haut, pour juger de la qualité d'une solution, devient impossible. C'est pourquoi, nous proposons un estimateur de f , noté \hat{f} , comme critère de comparaison entre les différentes solutions explorées par l'algorithme durant sa

recherche de la "meilleure" solution possible.

2.7.7 Solution "à priori" VS solution "à postérieur"

Avant de définir \hat{f} , l'estimateur de f que nous utilisons comme fonction d'évaluation des solutions explorées, il est nécessaire d'adapter la manière dont la solution est représentée pour tenir compte de l'incertitude entourant la survenue de détours en cours de route.

En effet, la représentation d'une tournée sous forme de vecteur contenant les points à visiter triés suivant l'ordre de passage du véhicule, telle que présentée dans la section 2.7.2, est en réalité une représentation incomplète d'une vraie tournée. En effet, seuls sont pris en compte avec cette représentation, les points de passage **obligatoires** (en d'autres termes, les points de livraison et les points de collecte). La notion d'incertitude qui accompagne les détours en cours de route n'y est pas prise en compte. On dira d'une solution qui ne prend pas en considération la survenue de détours que c'est une **solution à priori**, dans le sens où est elle construite **avant** le début de la tournée, à un moment où les informations concernant les détours ne sont pas encore connues.

De la même manière, on dira d'une solution qui prend en compte les éventuels détours pouvant survenir en cours de trajet que c'est une **solution à postérieur**, en ce sens qu'elle contient des informations ne pouvant être connues qu'**après** la fin de la tournée.

Ainsi, comme le montre le schéma 2.13, une solution à postérieur est une solution à priori à laquelle est associé un vecteur ω où ω_i est une variable booléenne telle que :

- $\omega_i = 1$ si un détour en centre de recyclage est requis immédiatement après la visite du point i ;
- $\omega_i = 0$ sinon.

Remarque 6. *Notons ici que si $\omega_i = 1$, alors i est forcément un point de collecte, car les détours en centre de recyclage ne se produisent, potentiellement, qu'après des collectes durant lesquelles du béton non consommé est récupéré.*

Remarque 7. *Rappelons également que ω_i est une variable dont la valeur n'est connue qu'au moment où le véhicule atteint le point i , car c'est uniquement à ce moment qu'est dévoilée l'information concernant la nécessité ou non d'un détour.*

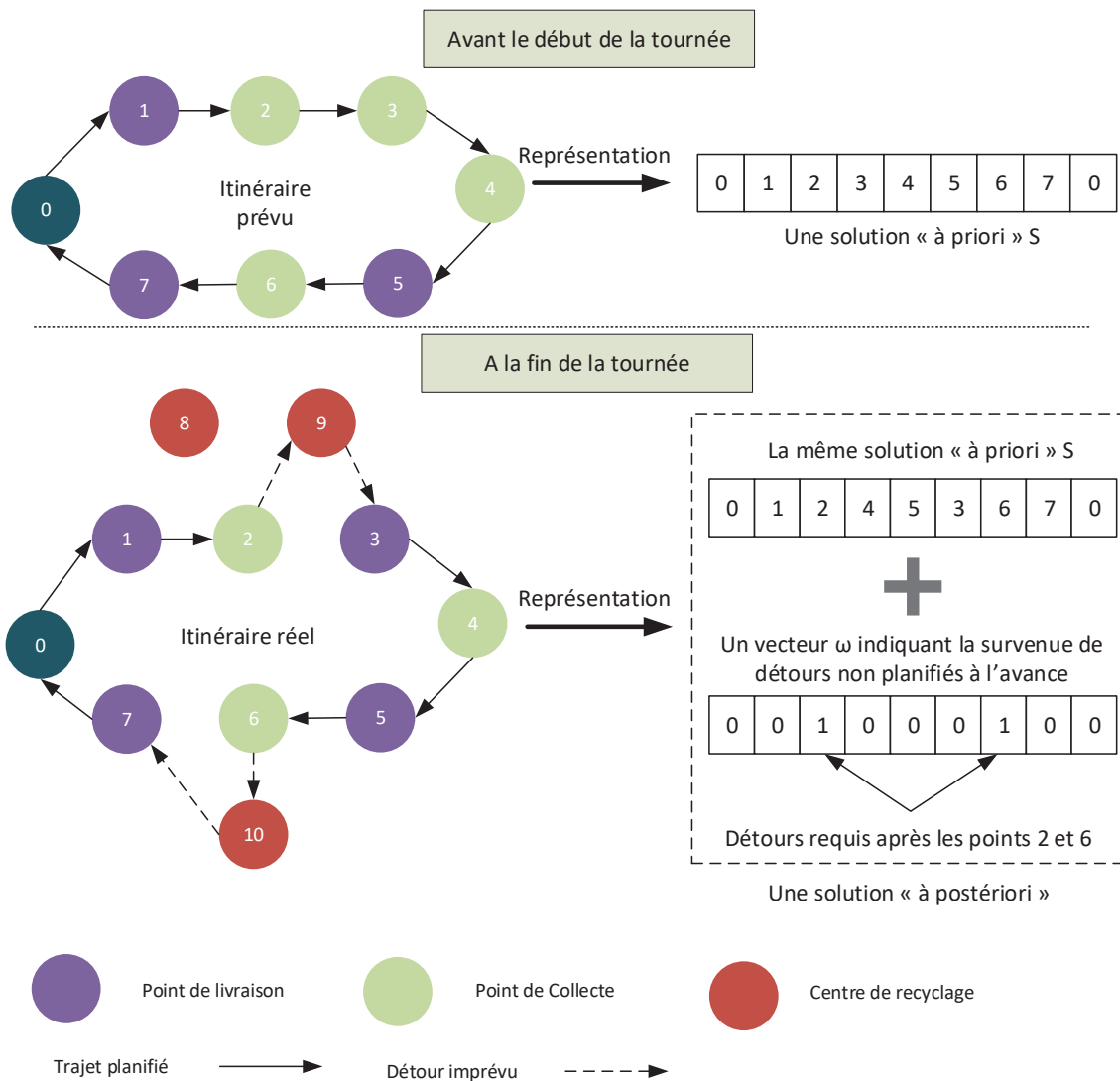


FIGURE 2.13 – Solution "à priori" VS solution "à postériori"

2.7.8 Estimateur de la fonction objectif

Supposons une solution "à priori" S , construite avant le début de la tournée. Pour calculer $f(S)$, il est nécessaire de connaître la solution "à postériori" (c'est-à-dire, la solution réelle, effective, incluant les détours effectués durant la tournée). En effet, pour une même solution "à priori" S , la valeur de $f(S)$ peut fortement varier en fonction du nombre de détours que le véhicule aura à effectuer en cours de route, et des endroits où ces détours seront nécessaires. La fonction f ne dépend donc pas uniquement de la solution S , mais également du vecteur ω qui représente le trajet réel constaté à la fin de la tournée (voir schéma 2.13).

Remarque 8. On notera désormais $f(S, \omega)$ la valeur de la fonction objectif pour une

solution S soumise à un scénario ω représentant les changements d'itinéraire imprévus.

ω n'étant pas connu à l'avance, calculer $f(S, \omega)$ devient impossible. A la place, on calculera une **valeur estimée** de $f(S, \omega)$ de la manière suivante :

$$\hat{f}_M(S) = \frac{1}{M} \sum_{r=1}^M f(S, \omega_r)$$

où :

- S est la solution dont on cherche à estimer la valeur de la fonction objectif ;
- ω_r est un scénario de tournée possible ;
- M est le nombre total de scénarios générés ;

Dit autrement, estimer le coût global d'une solution "à priori" S revient à calculer la moyenne des coûts de M solutions "à postériori" générées en associant la solution S à M scénarios de tournées possibles.

Les M vecteurs ω sont générés suivant un ensemble $P = \{p_i, i \in N_p\}$, où p_i est la probabilité qu'un détour en centre de recyclage soit requis immédiatement après la visite du point i ($p_i = 0 \forall i \notin N_p$). Ces probabilités sont calculées à partir de données historiques sur la fréquence à laquelle les clients surestiment la quantité de béton nécessaire pour effectuer leurs travaux. Surestimations qui, dans le cas de nos tournées, impliqueraient la récupération de béton non consommé, et donc, des détours en centres de recyclage.

2.7.9 Choix du centre de recyclage

Lorsqu'un passage en centre de recyclage est requis après la visite d'un point i , le véhicule doit se détourner de son itinéraire initial et choisir un des centres disponibles à proximité pour y acheminer les résidus de béton récupérés. La stratégie adoptée pour le choix du centre de recyclage est simple. Si le détour doit s'effectuer entre les visites des points i et j , alors, le centre k choisi est celui qui minimise le coût du détour effectué. En d'autres termes, k est choisi de telle sorte à minimiser $c_{i,k} + c_{k,j}$ (voir schéma 2.14).

2.7.10 Résumé

Nous avons exposé dans cette section une première approche pour construire des tournées de véhicule efficaces et répondant aux contraintes et spécificités énoncées dans la section 2.1.

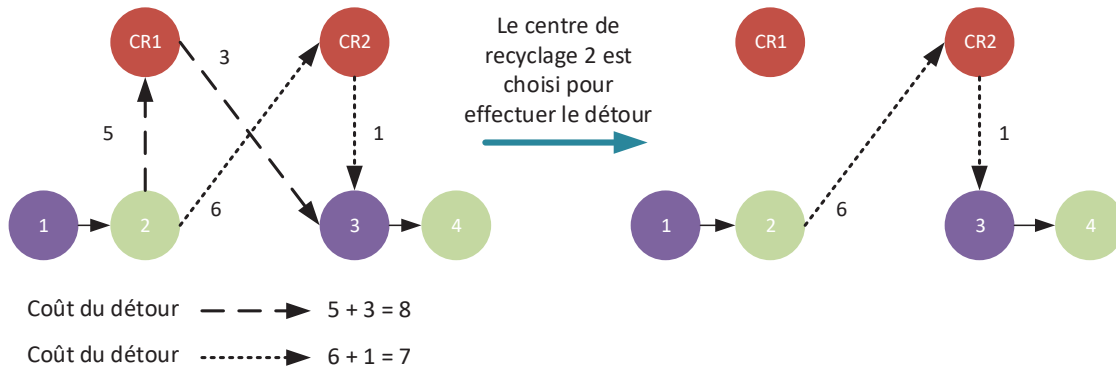


FIGURE 2.14 – Choix du centre de recyclage

Cette première approche est dite *statique et stochastique*. Elle est **statique** car l'ordre de passage du véhicule à chaque point est fixé avant le début de la tournée et ne change pas en cours de route. Seuls des détours en centres de recyclage peuvent, éventuellement, s'intercaler entre les points préalablement prévus, mais sans changer l'ordre dans lequel ils sont visités.

D'autre part, l'approche est dite **stochastique** car elle s'appuie sur des données incertaines, présentes sous forme de probabilités, pour anticiper les détours qui peuvent survenir en cours de trajet et minimiser, au préalable, leur impact sur le coût global de la tournée.

Concrètement, l'algorithme présenté plus haut se base sur une **recherche locale** que nous alimentons avec une **liste de solutions initiales** autour desquelles l'algorithme explore le **voisinage** à la recherche de solutions meilleures. Les solutions initiales sont générées à l'aide de différentes heuristiques (voir section 2.7.3), le but étant de diversifier les solutions pour couvrir un maximum d'espace de recherche, et éviter les optimum locaux.

La génération du voisinage d'une solution se fait à l'aide de l'heuristique *1-Shift* (voir section 2.7.4), quant à la fonction utilisée pour évaluer la qualité d'une solution, il s'agit du coût moyen que cette dernière engendre lorsqu'elle est soumise à différents scénarios de tournées possibles (un scénario représente le nombre de détours imprévus et les endroits auxquels ils surviennent durant la tournée).

Pendant la tournée, l'itinéraire généré par la recherche locale est suivi par le véhicule qui ne s'en éloigne que s'il est amené à passer en centre de recyclage, auquel cas, il se dirigera vers le centre engendrant le coût de détour le moins important parmi tous les centres

disponibles à proximité, avant de rejoindre l'itinéraire initialement prévu.

Le schéma 2.15 résume les principales étapes de l'approche statique et stochastique présentée dans cette section.

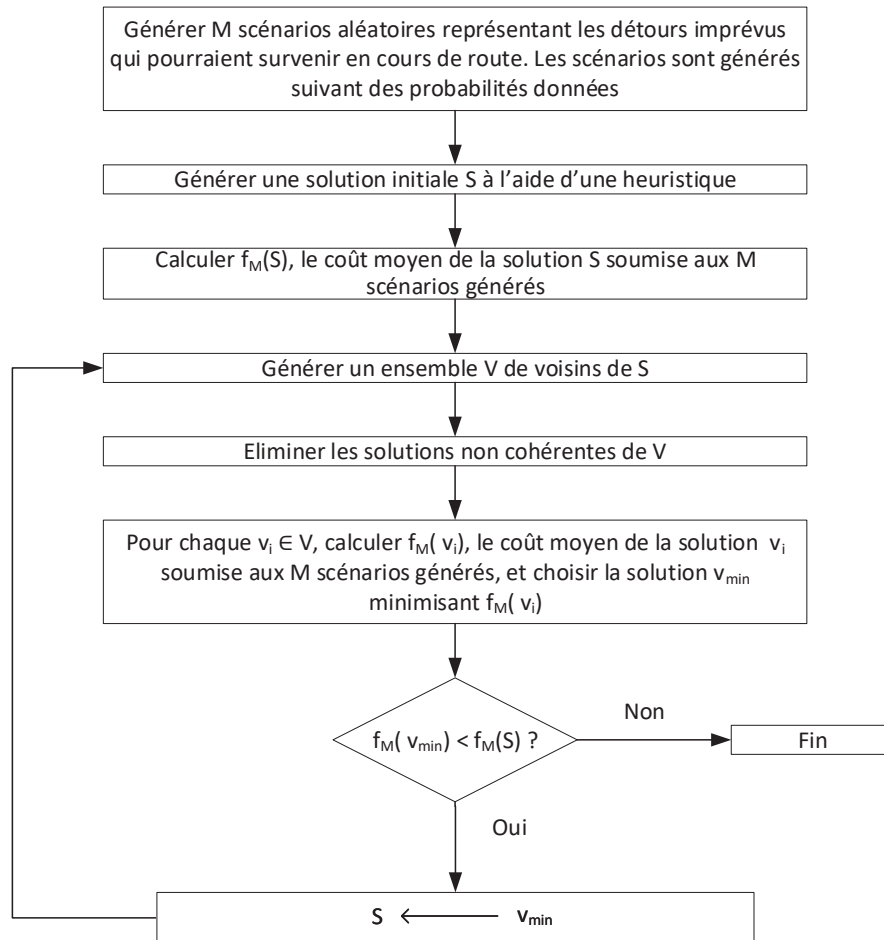


FIGURE 2.15 – Approche stochastique : vue globale

2.8 Approche dynamique

Après l'approche stochastique présentée dans la précédente section, nous nous penchons ici sur une approche dynamique pour construire des tournées robustes, efficaces, et respectant toutes les contraintes inhérentes à notre problématique.

2.8.1 Approche stochastique VS Approche dynamique

Contrairement à l'approche stochastique où les tournées sont planifiées en amont et ne changent pas en cours de route, l'approche dynamique consiste à construire une tournée, puis à la mettre à jour, en temps réel, au fur et à mesure de la découverte de nouvelles informations sur le déroulement de la tournée. Plus précisément, la nécessité pour le véhicule d'effectuer des détours pendant son trajet, le nombre de ces détours, et les endroits où ils surviennent sont autant d'informations qui ne sont pas en notre possession au préalable, et dont la connaissance, progressive, permet de remettre en question l'itinéraire calculé en amont de l'obtention de ces nouvelles données.

	Approche stochastique	Approche dynamique
Avant le début de la tournée	Une tournée est construite de sorte à minimiser le coût moyen engendré suivant différents scénarios de tournées possibles.	Un itinéraire est calculé de sorte à minimiser le coût de trajet global, dans le cas idéal (sans détours).
	L'algorithme simule plusieurs déroulements possibles de la tournée en exploitant des données stochastiques (probabilités de survenue de détours en cours de route).	Les éventuels détours en centres de recyclage ne sont pas pris en compte pour le calcul de la tournée.
Pendant la tournée	L'ordre de passage du véhicule par les différents points obligatoires n'est jamais remis en question.	L'ordre de passage du véhicule par les différents points obligatoires est remis en question à chaque fois qu'un détour est requis.
	En cas de nécessité d'un détour après la visite d'un point i , le centre de recyclage minimisant le coût du détour est choisi pour être visité entre i et le point suivant j .	L'algorithme qui a construit la tournée initiale (avant le début de la tournée) est relancé en cours de route, à chaque fois que le véhicule doit se détourner de l'itinéraire prévu, afin de recalculer le meilleur itinéraire possible compte tenu du nouveau contexte.

TABLE 2.1 – Approche stochastique VS Approche dynamique

2.8.2 Principe

De la même manière que pour l'approche présentée précédemment, l'algorithme que nous proposons ici est basé sur une recherche locale à laquelle nous soumettons un ensemble de solutions initiales construites à l'aide des mêmes heuristiques que celles présentées dans la section 2.7.3. Le mécanisme de génération du voisinage d'une solution est également le même que pour la première approche (voir section 2.7.4).

Notre approche passe par deux phases :

a) Avant le début de la tournée

L'algorithme est lancé avant que le véhicule ne quitte son dépôt, à un moment où on n'a pas encore de visibilité sur les éventuels futurs détours qui devront être effectués. Dans cette première phase, la notion d'évènement imprévu est ignorée. L'algorithme cherche une solution qui puisse satisfaire les contraintes énoncées et connues à l'avance, parmi lesquelles la contrainte de capacité du véhicule, sans essayer d'anticiper un quelconque évènement incertain.

Pour trouver la meilleure solution possible, l'algorithme s'appuie sur la fonction $f(S) = \sum_{i \in N_d \cup N_p} \sum_{j \in N_d \cup N_p} x_{i,j} c_{i,j}$ pour évaluer et comparer les solutions explorées tout au long du processus de recherche. $f(S)$ représente le coût global d'une solution S dans le meilleur des cas, en supposant qu'aucun évènement imprévu ne vienne dégrader la qualité de la tournée en cours de route.

La première phase s'achève donc avec la construction d'une solution cohérente et efficace, qui constituera la feuille de route initiale du véhicule.

b) Pendant la tournée

En cours de route, le véhicule, qui suit l'itinéraire initialement planifié, peut se trouver dans la nécessité de dévier du trajet qu'il est censé suivre, pour se diriger vers un centre de recyclage. À ce moment-là, l'algorithme de recherche qui a servi à générer la tournée initiale est exécuté à nouveau dans le but de trouver le meilleur chemin à emprunter, permettant d'abord de passer dans un des centres de recyclage à proximité, avant de visiter tous les points non encore atteints.

Ainsi, l'ordre de passage par chacun des points de livraison et de collecte est remis en question, et peut être modifié, en temps réel, à chaque fois que le contexte de la tournée change, ou, autrement dit, à chaque fois qu'on obtient une information (en l'occurrence, la nécessité d'un détour) que l'on ne possédait pas avant. C'est cette flexibilité de la tournée qui rend cette approche "dynamique".

L'algorithme 3 et le schéma 2.16 résument les principales étapes de cette approche.

Algorithme 3 : Approche dynamique

```

/* Phase 1 : avant le début de la tournée */
/* Une solution S est générée grâce à la recherche locale précédemment
   présentée, exécutée sur un ensemble de solutions initiales */
S ← LocalSearch(InitialSolutionSet) ;
/* Phase 2 : pendant la tournée */
/* Pour chaque point de livraison ou de collecte... */
for i ∈ S do
    /* Si un détour est requis après le point i */
    if ωi = 1 then
        BestSubSolution ← nil ;
        /* Pour chaque centre de recyclage à proximité... */
        for j ∈ Nr do
            /* Extraire les points de S qui n'ont pas encore été visités */
            Sj ← SubSolution(i + 1, n) ;
            /* Ajouter le centre de recyclage j à la liste des points non encore
               visités */
            Sj ← Sj.push(j) ;
            /* Lancer une recherche locale avec Sj comme solution initiale et j
               comme point de départ */
            Sj ← LocalSearch(Sj) ;
            /* Garder le chemin le moins coûteux parmi ceux proposés par la
               recherche locale pour chacun des centres de recyclage disponibles
               */
            if BestSubSolution = nil OR f(Sj) < f(BestSubSolution) then
                BestSubSolution ← Sj ;
        /* Remplacer le chemin initialement prévu pour les points non encore
           visités, par le chemin obtenu plus haut */
        Replace(SubSolution(i + 1, n), BestSubSolution) ;

```

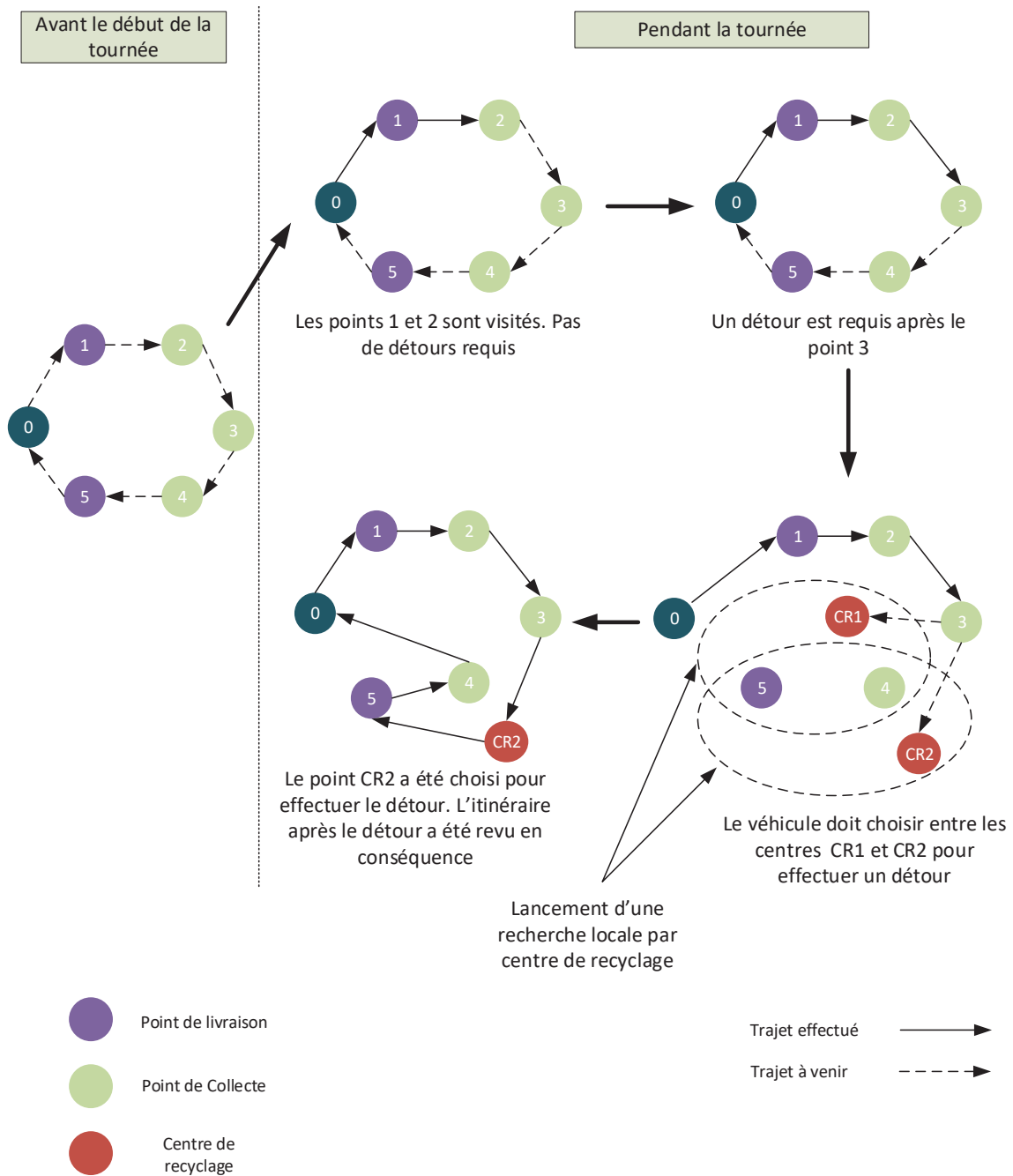


FIGURE 2.16 – Approche dynamique

2.9 Expérimentations

Nous présentons ici les tests auxquels les deux approches, stochastique et dynamique, ont été soumises.

2.9.1 Implémentation

Les algorithmes précédemment présentés ont été développés :

- en JAVA ;
- Sous LINUX (UBUNTU 18.04) ;
- sur une machine dotée d'un processeur AMD A10-7700K RADEON R7, 3.40 GHz et de 8 GB de RAM.

2.9.2 Jeu de données

Les expérimentations ont été menées sur des instances standards, largement utilisées dans la littérature. Ces instances ont été générées par [Gendreau et al., 1999], pour un problème similaire à celui abordé ici, le PDTSP (*Pickup & Delivery Travelling Salesman problem*). Chaque instance est représentée par :

- une matrice où l'élément (i, j) correspond au coût du trajet entre les points i et j , cela correspond aux valeurs $c_{i,j}$ présentées précédemment. Les éléments de la matrice respectent les propriétés mathématiques définissant une distance, à savoir, la symétrie, la séparation, et l'inégalité triangulaire ;
- un vecteur dont l'élément i correspondant à la demande du point i . Dans notre cas, il s'agira du nombre de bacs à livrer, ou à récupérer.

Les instances sont scindées en 6 classes selon le nombre de points à visiter : 25 points, 50 points, 75 points, 100 points, 150 points et 200 points. Chacune des classes est composée de 30 instances, ce qui fait en tout 180 instances.

Cela étant dit, les instances telles que proposées par [Gendreau et al., 1999] ne répondent que partiellement au besoin induit par notre problématique. En effet, les instances de GENDREAU, comme tout autre jeu de données à notre connaissance, ne prend pas en compte la notion bien particulière de passage éventuel par des points non prévus à l'avance. Ainsi, il nous a fallu les modifier pour définir des points (les centres de recyclage dans notre cas), dont la visite est conditionnée par des événements se produisant durant la tournée. Pour ce faire :

- les quatre premiers points de chaque instance sont exclus de la liste des points obligatoirement visités durant la tournée, et sont considérés comme des centres de recyclage, éventuellement visités en cours de route ;
- un scénario "réel" est associé à chacune des instances. Autrement dit, pour chaque

instance, nous générons un vecteur décrivant le déroulement réel de la tournée correspondante. Il indique les endroits où des détours sont requis en cours de trajet, et c'est par rapport à ce scénario considéré comme "réel" que les solutions trouvées par nos approches sont évaluées.

Il est à noter que les scénarios réels évoqués ci-dessus sont générés en prenant en compte, pour chaque point visité i , la probabilité p_i qu'un détour soit requis immédiatement après. Ainsi, pour tester nos approches, nous considérons différents types de scénarios, correspondant à des tournées plus ou moins sujettes à des détours imprévus. Nous soumettons alors les solutions obtenues par nos algorithmes à des scénarios où les détours sont très rares (où la probabilité de survenue d'un détour $p_i = 0,1$ pour chaque point de collecte visité), et à des scénarios où les détours sont très fréquents ($p_i = 0,9$ pour chaque point de collecte visité), en passant par $p_i = 0,3$, $p_i = 0,5$, et $p_i = 0,7$.

2.9.3 Résultats

Le tableau 2.2 montre les résultats obtenus avec l'approche stochastique (voir section 2.7), pour chacune des classes d'instances du benchmark de [Gendreau et al., 1999].

Pour chaque instance, le coût global de la solution trouvée par l'algorithme est calculé, en prenant en compte les détours requis pendant la tournée (indiqués par le scénario "réel" associé à l'instance considérée). Le résultat obtenu pour une classe donnée est la moyenne des coûts obtenus pour chacune des instances qui composent cette classe.

En outre, l'approche est testée pour différentes valeurs de p , la probabilité de survenue de détours en cours de route, et pour différentes valeurs de M , la taille de l'échantillon de scénarios "possibles" utilisé par l'algorithme pour évaluer les solutions "à priori".

Remarque 9. *Notons que pour $M = 0$, l'approche stochastique n'utilise aucun échantillon de scénarios "possibles" pour anticiper et évaluer l'impact d'éventuels détours durant la tournée. L'approche est alors équivalente à une simple recherche locale où l'incertitude entourant l'itinéraire réellement effectué n'est jamais prise en compte.*

Remarque 10. *La taille des voisinages générés par nos algorithmes est fixée à 200. Ce nombre offre un compromis intéressant entre qualité des solutions obtenues et vitesse d'exécution des algorithmes.*

À la lumière des résultats présentés dans le tableau 2.2, nous notons d'abord que,

Probabilité	Nombre de points	Taille de l'échantillon M			
		0	25	50	100
$p = 0.1$	25	544,26	555.9	551,5	565,4
	50	843,33	895	893.1	860.6
	75	1121,1	1119	1138,66	1111,5
	100	1414,56	1432,5	1494	1457,5
	150	2007,07	2017	2013	1993,3
	200	2603,73	2601.1	2496	2489,16
$p = 0.3$	25	618,5	596	574,26	577,3
	50	1020,03	1018,2	1051,3	1001,1
	75	1446,9	1474,4	1398,12	1392,8
	100	1874,26	1862,2	1813.7	1841,1
	150	2673,83	2641,14	2594	2569,4
	200	3592,26	3617,14	3504,2	3495,6
$p = 0.5$	25	706,66	688,3	681.13	680,6
	50	1281,33	1210,15	1187	1161,7
	75	1848,86	1817,21	1832,2	1804,4
	100	2233,96	2214,1	2157	2149,4
	150	3416,86	3411,14	3378,9	3386.6
	200	4445,53	4431,37	4376,1	4348,4
$p = 0.7$	25	782,06	771	734,3	746,2
	50	1480,83	1535,2	1457,5	1447,3
	75	2172	2267,3	2169	2125,2
	100	2666,5	2517,3	2500,1	2491,7
	150	4047,16	4006,7	4011,6	3992,8
	200	5490,2	5397,3	5325,8	5332,3
$p = 0.9$	25	867,8	804,4	786,4	799,1
	50	1682,06	1633,9	1526,1	1589,1
	75	2472	2480,4	2366,5	2306,2
	100	3108,8	3116,5	2915,2	2903
	150	4741,8	4886,3	4605,36	4552,52
	200	6457,8	6284,8	6176,6	6001,1

TABLE 2.2 – Coût global moyen des solutions obtenues avec l'approche stochastique sur les instances de [Gendreau et al., 1999]

globalement, le coût des solutions trouvées diminue au fur et à mesure que la taille de l'échantillon de scénarios utilisé par l'algorithme grandit. Le seul cas de figure où ce constat n'est pas vérifié est celui dans lequel les détours en cours de route sont très improbables ($p = 0, 1$).

Ainsi, il apparaît que notre approche basée sur l'anticipation d'éventuels futurs détours, et l'estimation de l'impact qu'ils peuvent avoir sur le coût des solutions, est bien plus performante qu'un algorithme classique qui se contenterait de calculer des tournées sans prendre en considération des données stochastiques permettant de prévoir les possibles coûts additionnels liés aux détours imprévus (c'est le cas de notre approche lorsque $M = 0$).

L'approche stochastique ne permet pas de trouver les meilleures solutions dans le cas idéal où des événements imprévus n'ont jamais lieu, mais elle permet de trouver les solutions qui s'adaptent le mieux aux imprévus, et dont le coût est le moins sensible aux changements d'itinéraire en cours de route.

Néanmoins, une telle approche n'est pertinente que lorsque les détours au cours d'une tournée sont suffisamment nombreux pour que leur coût ne soit pas négligeable face au coût global de la tournée, ce qui explique pourquoi notre approche stochastique obtient de moins bons résultats lorsque $p = 0, 1$, car c'est un cas de figure où les détours sont tellement rares (surtout pour les instances avec 25 ou 50 points à visiter) que l'algorithme ne parvient pas à les anticiper correctement. Dans ce cas-là, négliger les possibles futurs détours s'avère plus efficace, cela revient à lancer l'algorithme avec $M = 0$.

Par ailleurs, nous notons également que le coût moyen des solutions obtenues pour chacune des classes d'instances augmente à mesure que le paramètre p augmente. En effet, plus la valeur de p est élevée, plus le risque de devoir effectuer des détours est élevé. La croissance du coût global des solutions est donc liée à la part importante des détours dans le coût global des solutions.

Cela étant dit, nous remarquons que le taux d'augmentation du coût moyen des solutions évoqué ci-dessus est moins fort lorsque la valeur du paramètre M est élevée. En effet, le tableau 2.3 compare le taux d'augmentation du coût moyen des solutions trouvées lorsque $p = 0, 9$ par rapport au cas où $p = 0, 1$, pour $M = 0$ et $M = 100$. Dit autrement, le tableau 2.3 montre à quel point les coûts peuvent varier lorsque des tournées sont soumises, d'abord à une très faible ($p = 0, 1$), puis à une très forte ($p = 0, 9$) probabilité

de survenue de détours.

Nombre de Points considérés	$M = 0$	$M = 100$
25	+59,42 %	+41,33 %
50	+99,45 %	+84,65 %
75	+120,49 %	+107,48 %
100	+119,77 %	+99,17 %
150	+136,25 %	+128,39 %
200	+148,02 %	+141,08 %

TABLE 2.3 – Taux d’augmentation du coût moyen des solutions obtenues lorsque la probabilité de survenue de détours $p = 0,9$, par rapport au cas où $p = 0,1$

Il apparait que lorsqu’elles sont générées avec $M = 100$, les solutions sont non seulement de meilleure qualité que celles générées avec $M = 0$, mais également plus "résistantes" à la multiplication du nombre de détours imprévus dans le sens où leur coût augmente moins vite que lorsque les imprévus sont complètement ignorés au moment de la construction des solutions. Le fait d’anticiper les détours permet donc d’avoir des solutions plus robustes et moins sensibles aux changements de dernière minute.

Le graphique 2.17 montre la part du coût moyen des solutions obtenues qui est due aux détours.

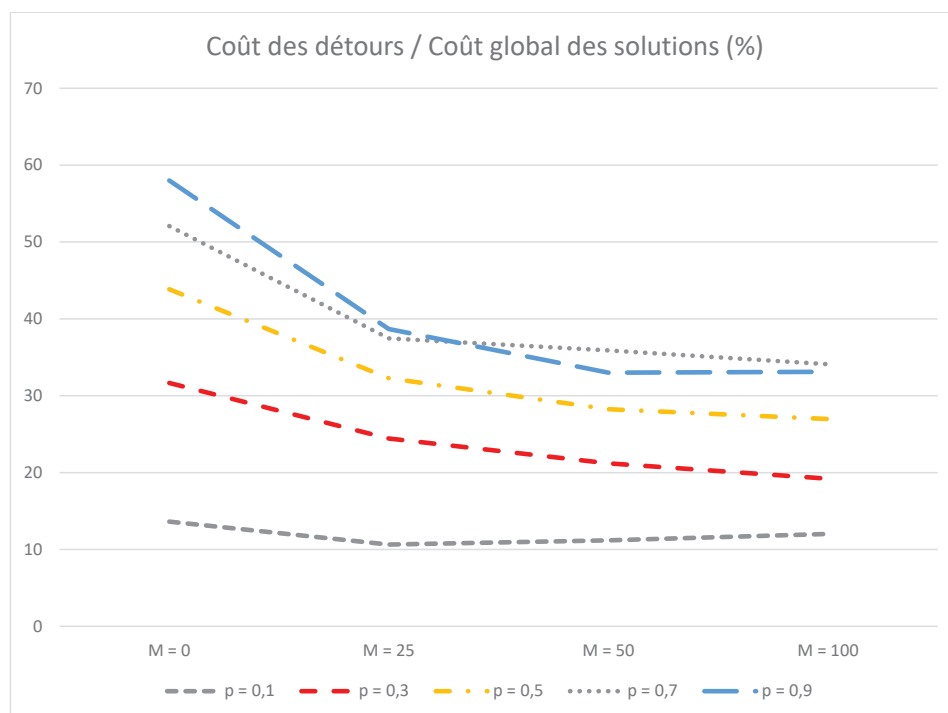


FIGURE 2.17 – Part des détours dans le coût moyen global des solutions trouvées

Les résultats montrent une baisse significative de la part des coûts due aux détours lorsque le nombre de scénarios utilisés pour les anticiper augmente. Ainsi, à partir de $p = 0,3$, nous constatons une baisse d'au moins 10% du pourcentage des coûts causé par les détours. Ce pourcentage frôle même les 30% (il passe de 59% pour $M = 0$ à 32% pour $M = 100$) lorsque $p = 0,9$. Ainsi, lorsque les détours sont très probables, l'intérêt de l'approche stochastique est encore plus grand, car plus les détours sont nombreux, plus leur impact peut être important, et plus on aura intérêt à exploiter une approche dont l'avantage est de minimiser l'impact des imprévus en cours route.

Notons tout de même que la part des détours dans le coût global des solutions reste stable lorsque $p = 0,1$. En effet, dans ce cas, et comme évoqué plus haut, les détours sont tellement peu nombreux que leur impact sur les tournées est minime. Il est donc difficile de les anticiper correctement.

Le tableau 2.4 compare les approches stochastique et dynamique.

Remarque 11. *Pour la colonne représentant l'approche stochastique, les chiffres affichés correspondent aux résultats obtenus lorsque $M = 100$.*

La comparaison penche clairement en faveur de l'approche dynamique, et ce quel que soit la valeur de p considérée.

Ainsi, l'approche dynamique obtient de meilleurs résultats pour 80% des instances testées. Mieux encore, elle obtient de meilleurs résultats pour 100% des instances dont le nombre de points est supérieur ou égal à 75. Cela est dû à la flexibilité de cette approche qui, contrairement à l'algorithme stochastique, ne se contente pas d'anticiper à priori les futurs détours pour faire en sorte de réduire leur impact sur la qualité des tournées, mais recalcule dynamiquement, pendant la tournée, le meilleur itinéraire à emprunter après chaque détour.

Conclusion

Dans ce chapitre, nous avons présenté le premier schéma logistique proposé pour optimiser le processus de livraison de béton en petites quantités.

Le scénario étudié fait appel à un camion grue effectuant une tournée durant laquelle il doit livrer des bacs remplis de béton à des clients qui en ont fait la demande, tout en col-

Probabilité	Nombre de points considérés	Approche stochastique	Approche dynamique
$p = 0.1$	25	565,4	533,16
	50	860,6	803
	75	1111,5	1043,6
	100	1457,5	1218,36
	150	1993,3	1613,86
	200	2489,16	2091,16
$p = 0.3$	25	577,3	619,6
	50	1001,1	1051,5
	75	1392,8	1310,16
	100	1841,1	1686,46
	150	2569,4	2381,63
	200	3495,6	3004,4
$p = 0.5$	25	680,6	699,43
	50	1161,7	1245,4
	75	1804,4	1678,7
	100	2149,4	2146
	150	3386,6	3017,73
	200	4348,4	4050,2
$p = 0.7$	25	746,2	766,83
	50	1447,3	1414,93
	75	2125,2	2038,33
	100	2491,7	2486,8
	150	3992,8	3709,5
	200	5332,3	5013,5
$p = 0.9$	25	799,1	864,16
	50	1589,1	1578,4
	75	2306,2	2287,23
	100	2903	2887,73
	150	4552,52	4472,4
	200	6001,1	5722,7

TABLE 2.4 – Coût global moyen des solutions obtenues avec l’approche stochastique et l’approche dynamique sur les instances de [Gendreau et al., 1999]

lectant des bacs vides chez d’autres clients (ayant été livrés lors d’une précédente tournée). Ce faisant, le véhicule ne peut transporter plus de bacs que sa capacité de charge ne le permet, et doit respecter une contrainte inhérente au béton prêt à l’emploi, la contrainte de recyclage.

Cette dernière stipule que si un client ne consomme pas la totalité du béton qui lui est livré dans les bacs, alors, le véhicule, après avoir collecté ces bacs non vides, se retrouve dans l’obligation de les acheminer en priorité vers un centre de recyclage avant de pouvoir poursuivre sa tournée. L’état dans lequel se trouveront les bacs au moment de leur collecte n’étant pas connu à l’avance, le déroulement de la tournée se retrouve soumis à

l'incertitude entourant le survenue de détours en centres de recyclage.

Le problème posé par ce schéma logistique est un problème de tournées de véhicules avec livraison et collecte. Il s'agit d'un sujet souvent étudié dans la littérature, mais rarement dans un contexte d'incertitude, comme c'est le cas pour notre travail.

Après avoir formalisé la problème et clairement défini ses contraintes, nous avons proposé deux approches pour y répondre. La première, l'approche stochastique, essaye d'anticiper les imprévus qui peuvent survenir en cours de route pour minimiser leur impact sur la qualité des tournées générées. La seconde quant à elle, l'approche dynamique, permet de réadapter en temps réel l'itinéraire à emprunter au cours d'une tournée, et ce à chaque fois qu'un détour est effectué par le véhicule.

Les deux approches ont été testées sur des datasets standards, largement utilisés dans la littérature, et les résultats obtenus sont satisfaisants, notamment en ce qui concerne la capacité des algorithmes à minimiser l'impact des détours sur le coût global des tournées.

Schéma logistique 2 : Tournées avec deux véhicules synchronisés

Introduction

Dans le précédent chapitre, nous avons décrit un premier schéma logistique pour la livraison de béton en petites quantités. Il s'agissait, pour un **camion grue**, de déposer des bacs remplis de béton chez les nouveaux clients qui en font la demande, et de collecter les bacs vides chez les clients de la veille, le tout, au cours d'une même tournée.

Dans ce chapitre, nous nous intéressons à un second schéma logistique impliquant deux types de véhicules, **un camion toupie, et un véhicule utilitaire léger**.

Nous présenterons dans ce qui suit :

- le fonctionnement, les contraintes, et les objectifs de ce second schéma logistique ;
- les différences notables par rapport au premier schéma présenté ;
- un état de l'art sur les travaux de recherche proches de la problématique que nous traitons ici ;
- notre approche pour traiter cette problématique ;
- les résultats des tests effectués sur les algorithmes proposés.

3.1 Description du schéma logistique

Le scénario de livraison étudié ici fait appel à deux types de véhicules. Un véhicule utilitaire léger pour livrer et collecter des **bacs à béton vides**, et un camion toupie pour

remplir les bacs fraîchement déposés chez les clients.

Le fonctionnement de ce schéma est détaillé ci-dessous, d'abord du point de vue du client, puis de celui de l'entreprise.

3.1.1 Point de vue client

Lorsqu'un client est éligible à la livraison de béton en bacs, qu'il passe commande, et qu'il indique son adresse et la date de livraison souhaitée, les étapes suivantes seront nécessaires à la livraison de son béton suivant le schéma logistique considéré dans ce chapitre :

1. le jour de la livraison, un véhicule utilitaire se présente à l'adresse indiquée par le client pour y déposer un ou plusieurs bacs à béton **vides** (le nombre de bacs déposés dépend du volume commandé par le client, sachant qu'un bac peut contenir jusqu'à 500 litres de BPE). Une fois les bacs vides déposés, le véhicule quitte les lieux ;
2. dans la foulée du passage du véhicule utilitaire, un camion toupie arrive chez le client (qui vient donc de recevoir des bacs vides prêts à être remplis). Le camion toupie remplit les bacs avec la quantité préalablement commandée par le client, et s'en va à son tour ;
3. une fois les bacs reçus et remplis, le client a toute la journée pour effectuer ses travaux en piochant du béton frais, à chaque fois qu'il en a besoin, dans les bacs qui lui ont été livrés ;
4. le lendemain, le véhicule utilitaire qui avait livré les bacs vides la veille repasse les récupérer, à nouveau vides (voir figure 3.1).

3.1.2 Point de vue de l'entreprise

Pour l'entreprise, les événements s'enchaînent de la manière suivante :

- un véhicule utilitaire quitte son dépôt et entame sa tournée chargé de bacs à béton vides, destinés à être livrés à un ensemble de clients ayant passé commande quelques jours auparavant ;
- le même jour, à peu près simultanément, un camion toupie quitte sa centrale à béton pour entamer sa tournée, chargé d'une quantité de BPE suffisante pour

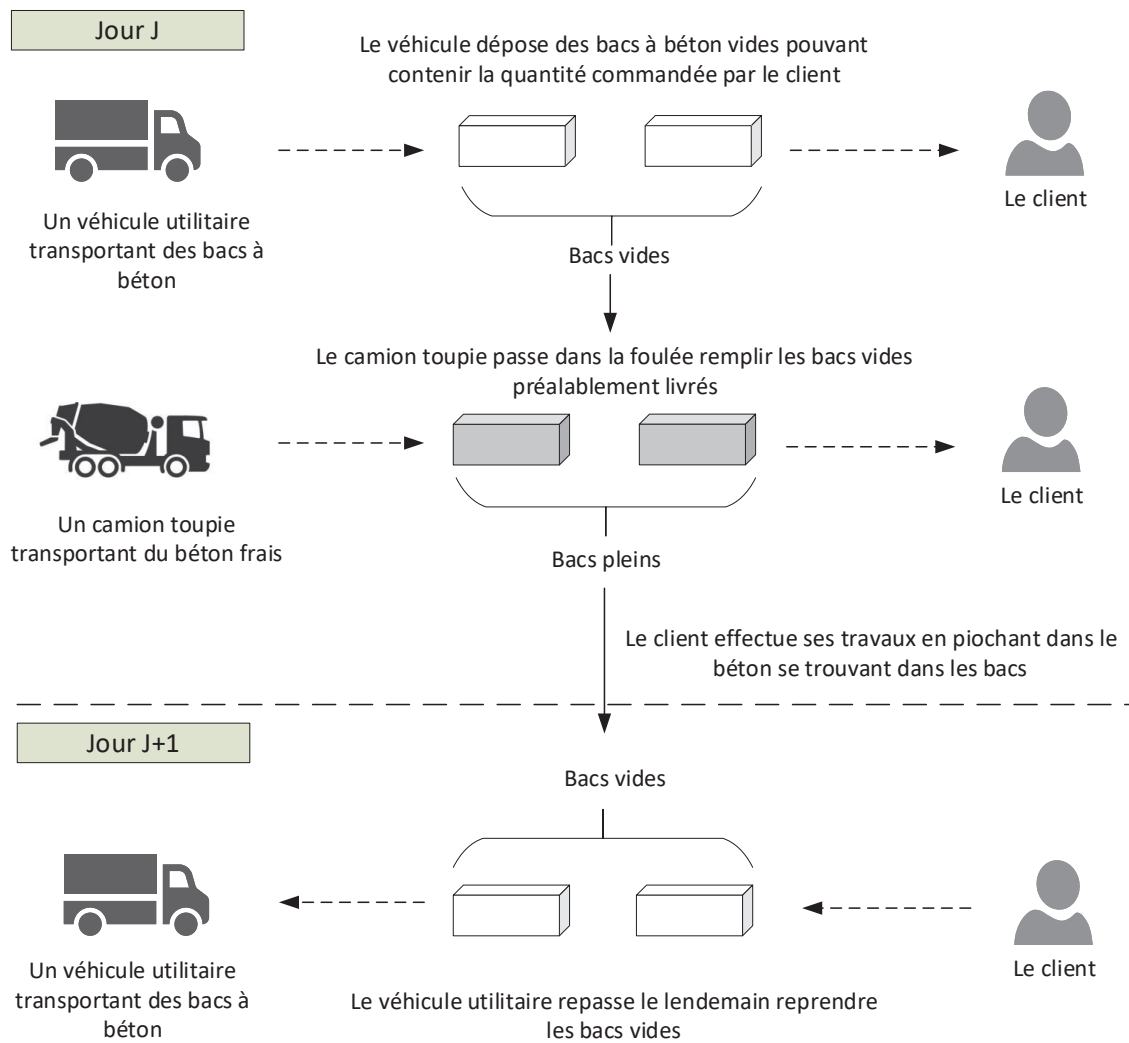


FIGURE 3.1 – Le schéma logistique n°2 vu par le client

- répondre au besoin des clients visités durant le parcours ;
- au cours de sa tournée, le véhicule utilitaire suit un itinéraire lui permettant de déposer des bacs chez les clients du jour, et d'en collecter chez les clients de la veille ;
 - au cours de sa tournée, le camion toupie suit un parcours permettant de visiter tous les clients du jour (ceux qui reçoivent, en parallèle, la visite du véhicule utilitaire qui livre les bacs). La toupie peut également se retrouver dans l'obligation de se réapprovisionner en béton en cours de route si la quantité transportée au départ ne suffit pas à satisfaire toutes les demandes des clients ;
 - pour chaque client devant recevoir la visite des deux véhicules (utilitaire et camion toupie), la visite du véhicule utilitaire doit toujours précéder celle de la toupie.

En effet, le rôle de la toupie est de remplir les bacs préalablement livrés par le véhicule léger. Ainsi, en arrivant avant que les bacs ne soient livrés, la toupie se retrouve dans l'obligation d'attendre chez le client le passage du véhicule utilitaire pour pouvoir remplir les bacs, ce qui n'est pas acceptable, d'une part parce que l'immobilisation du camion coûte cher, en temps et en argent, mais également parce que le camion toupie est un véhicule lourd dont le stationnement prolongé n'est pas toujours possible en ville ;

- le véhicule utilitaire revient à son point de départ (son dépôt) une fois qu'il a livré tous les bacs destinés aux clients du jour, et qu'il a récupéré tous ceux se trouvant chez les clients de la veille ;
- le camion toupie revient à son point de départ (son dépôt) une fois qu'il a rempli tous les bacs livrés par le véhicule utilitaire (voir figure 3.2).

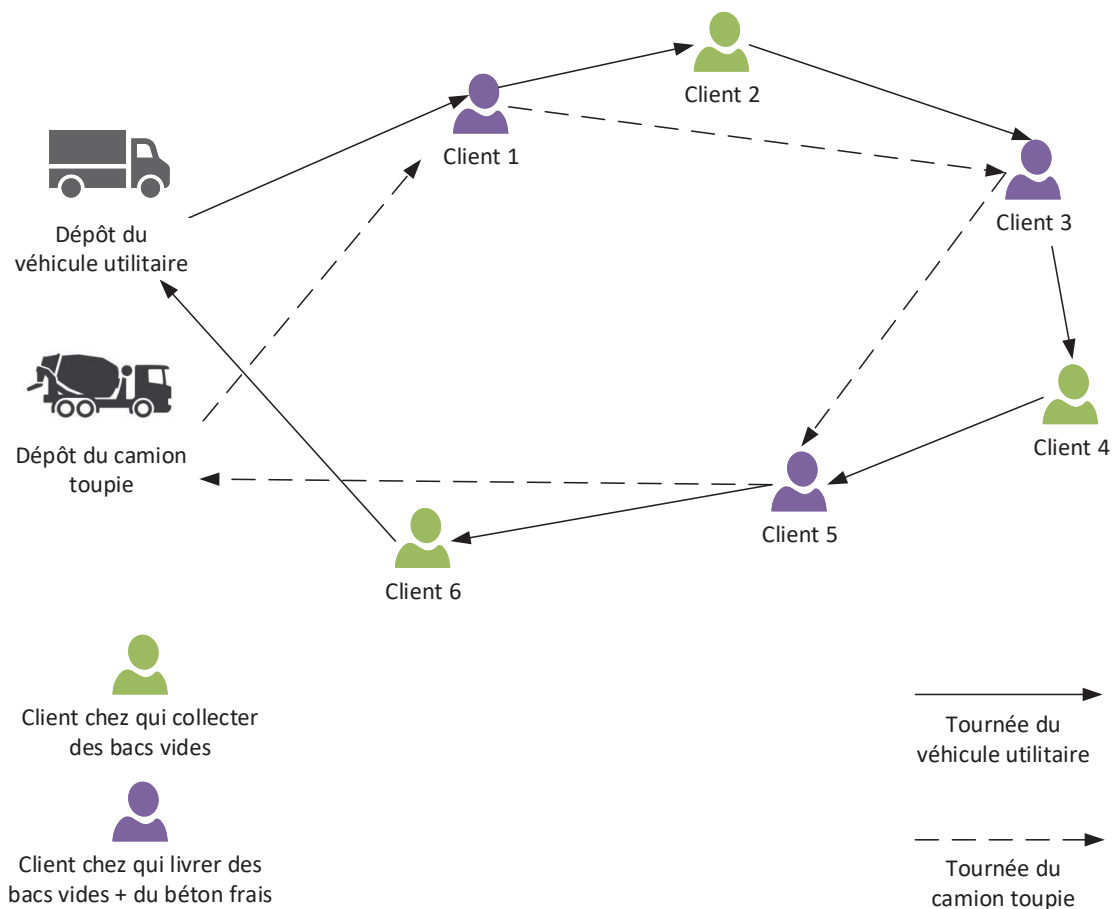


FIGURE 3.2 – Le schéma logistique n° 2 vu par l'entreprise

3.2 Schéma 1 VS Schéma 2

Le schéma logistique décrit ci-dessous et celui présenté dans le chapitre 2 visent tous les deux le même objectif : proposer un mode de livraison de BPE plus adapté aux petites quantités. Néanmoins, les deux schémas impliquent des types de véhicules différents, des contraintes différentes, et donc des déroulements différents. Pour rappel, le premier schéma logistique étudié fait appel à un véhicule lourd (un camion grue), capable de transporter un certain nombre de bacs remplis de béton prêt à l'emploi (le BPE étant un matériau très lourd, il est nécessaire d'avoir recours à un véhicule imposant pour supporter le poids de la charge). Le véhicule effectue ainsi sa tournée de livraison de ces bacs remplis de béton, et en profite pour récupérer, au cours de la même tournée, les bacs livrés la veille, vidés entre-temps par les clients. Ce schéma logistique requiert donc des tournées avec livraisons et collectes, impliquant un seul véhicule.

Cette façon de faire présente l'avantage de n'avoir recours qu'à un seul type de véhicules, ce qui permet plus de simplicité dans la gestion des flottes allouées à la livraison de béton (acquisition et maintenance des véhicules, recrutement de chauffeurs...), et évite de devoir traiter des problèmes de synchronisation entre différents types de véhicules...

Cependant, le principal inconvénient de ce mode de livraison est le fait de devoir transporter des bacs pleins. En effet, livrer du béton très lourd d'une part, et très encombrant car disposé dans des bacs volumineux, rend difficile la possibilité de répondre aux demandes d'un nombre important de clients au cours d'une même tournée, ce qui oblige à en organiser plusieurs, ce qui a pour conséquences de réduire la rentabilité du modèle et d'augmenter l'impact carbone des livraisons.

Ce constat nous a poussés à imaginer un deuxième schéma de livraison, qui gommerait les défauts du premier. C'est ainsi qu'on a étudié le 2ème schéma logistique, présenté plus haut dans ce chapitre.

Ici, le principal avantage est de ne jamais avoir à transporter des bacs pleins. Ceux-ci sont livrés et récupérés vides chez les clients. Les bacs vides étant légers et empilables, un véhicule utilitaire simple, bien moins encombrant, moins coûteux, et moins polluant que le camion grue du premier schéma, est suffisant pour transporter un très grand nombre de bacs. La livraison du béton est assurée par un camion toupie qui, lui aussi, peut transporter des quantités plus conséquentes que celles supportées par le camion grue du précédent schéma. On se retrouve donc avec un modèle qui contourne le problème de capacité de

transport limitée liée au caractère encombrant des bacs pleins.

Néanmoins, cette augmentation de la capacité de livraison au cours d'une tournée passe par le recours à deux différents types de véhicules (utilitaire et toupie). Bien que le bilan carbone d'une tournée suivant le second schéma reste comparable à celui d'une tournée adoptant le schéma 1 (l'impact de l'utilitaire étant négligeable par rapport aux autres types de véhicules), il n'en reste pas moins que ce deuxième schéma introduit plus de complexité dans l'organisation des tournées (à cause de la nécessité de synchroniser les visites chez les clients, notamment parce que la toupie doit toujours arriver chez le client après le véhicule utilitaire...).

Pour résumer donc, le schéma 1 est :

- plus simple à mettre en œuvre car il ne fait appel qu'à un seul véhicule qui se charge de l'intégralité du travail (collecte et livraison de bacs, livraison de béton) ;
- limité en termes de nombre de commandes pouvant être traitées au cours d'une même tournée.

Alors que le schéma 2 est :

- plus complexe dans sa mise en œuvre (recours à différents types de véhicules, nécessité de synchroniser les visites des différents véhicules chez les clients)
- plus rentable dans le sens où une tournée permet de traiter un plus grand nombre de demandes de clients.

3.3 Tournées de véhicules synchronisées

Comme évoqué dans le chapitre précédent, les problèmes de tournées de véhicules (VRP) suscitent, depuis très longtemps, un grand intérêt auprès des chercheurs, qui les ont étudiés de diverses manières et sous diverses formes en prenant notamment en compte un large éventail d'hypothèses et de contraintes reflétant souvent des situations rencontrées dans la pratique. Cela étant dit, il existe des variantes impliquant certaines contraintes ou certains objectifs particuliers qui n'ont été que très peu abordées dans la littérature comparativement aux VRP "classiques".

Les VRP avec contraintes de synchronisation font partie de ces variantes pour lesquelles des travaux ont commencé à émerger relativement récemment et qui sont à l'heure actuelle assez peu étudiées.

Un VRP avec contraintes de synchronisation est un VRP impliquant au moins 2 véhicules. Ceux-ci doivent, comme pour un VRP classique, visiter un certain nombre de nœuds pour y effectuer des tâches précises (livraison d'un bien, collecte...). La particularité des VRP avec contraintes de synchronisation est qu'au moins une partie des nœuds considérés requiert les visites synchronisées de plus d'un véhicule. Ainsi, si on considère un problème à 2 véhicules, un nœud i peut nécessiter les passages des deux véhicules, simultanément, l'un après l'autre en suivant un ordre prédéfini, ou dans n'importe quel ordre.

Le besoin d'effectuer des visites synchronisées chez des "clients" se retrouve dans de nombreuses applications concrètes. Par exemple, [Eveborn et al., 2006] se penchent sur un cas d'étude réel impliquant une structure sanitaire en Suède qui s'occupe de personnes âgées ou en situation de handicap. Les personnes faisant appel aux services de cet organisme peuvent requérir différentes prestations à domicile (soins, ménage...), accomplies par différents employés de l'organisme. Il faut alors prévoir des tournées pour chacun des employés en s'assurant que tous les "clients" soient visités par chacun des employés auxquels ils ont demandé des prestations. D'autres contraintes sont également étudiées dans ce papier comme le fait d'affecter, si possible, certains employés spécifiques à certains clients (il peut être apprécié de certains clients, pour une prestation donnée, d'avoir affaire au même employé à chaque fois...). Une heuristique a été adoptée pour résoudre le problème. Dans le même contexte pratique, [Rabeh et al., 2011] proposent un programme linéaire en nombres entiers pour des tournées synchronisées impliquant deux soignants devant rendre visite à des patients dans un ordre prédéfini.

Pour un problème d'allocation de tâches à des équipes de travailleurs devant, sur certaines tâches, travailler en collaboration (de façon synchronisée), [Dohn et al., 2009] proposent une approche exacte basée sur un *Branch & Price*. [Dohn et al., 2011] étudient quant à eux la contrainte de synchronisation en rajoutant des conditions sur les durées de temps minimale ou maximale entre deux visites chez un client, et proposent également une approche exacte basée sur un *Branch & Price*. De leur côté [Bredström and Rönnqvist, 2008] proposent un modèle mathématique ainsi qu'une heuristique pour résoudre des problèmes de tailles plus importantes.

Un autre exemple de problèmes réels impliquant des VRP avec contraintes de synchronisation a été présenté par [Salazar-Aguilar et al., 2012]. Partant d'un cas concret au Canada, le problème qu'ils traitent consiste à générer des tournées pour une flotte de

véhicules dans le but de déneiger des routes après une tempête de neige. Un déneigement simultané, par plusieurs véhicules, est alors requis pour certaines routes ou portions de routes disposant de plusieurs voies. Pour ce problème, ils proposent un PLNE ainsi qu'une heuristique basée sur une recherche locale. Parmi les travaux adoptant des méthodes de recherche de voisinage, on peut également citer [Liu et al., 2019] et [Affi et al., 2016]. Ces derniers ont appliqué un recuit simulé à un VRP avec contraintes de synchronisation et fenêtres de temps durant lesquelles les clients peuvent être visités. Les résultats obtenus sont jugés très satisfaisants.

Contrairement aux VRP vus dans le chapitre précédent, pour lesquels le nombre conséquent de travaux dans la littérature a fait émerger différentes façons d'approcher les problèmes, ici, pour les VRP avec contraintes de synchronisation, les travaux semblent assez peu nombreux pour permettre de voir se dégager des tendances, des idées dominantes, ou des familles d'approches. Certains travaux adoptent des méthodes de résolution exactes quand d'autres essaient des méthodes approchées, notamment lorsque les tests sont effectués sur des instances réelles. Les métaheuristiques, qui pourraient être intéressantes, notamment pour des applications réelles, sont, à notre connaissance, assez rarement étudiées pour ce genre de VRP.

Le problème posé par le schéma logistique étudié dans ce chapitre est un VRP à 2 véhicules où un ensemble de nœuds N doit être visité par le premier véhicule tandis qu'un sous-ensemble de N doit être visité par les deux véhicules, le premier devant toujours précéder le second. Le but visé est la minimisation du temps de trajet total effectué par les deux véhicules. Nous exposons dans ce qui suit deux approches à exploration de voisinage pour résoudre le problème. Dans la première, l'approche à optimisation séquentielle, il s'agira de construire les deux tournées séparément, l'une après l'autre. Pour la seconde, l'approche à optimisation simultanée, les deux tournées seront générées en même temps, l'une pouvant s'adapter à l'autre.

3.4 Formalisation du problème

Le schéma logistique décrit dans la section 3.1 implique la construction de tournées faisant appel à 2 véhicules circulant simultanément, et de façon synchronisée.

Afin de définir ce problème de façon formelle, nous notons $V1$ et $V2$ les deux véhicules participant à une tournée tels que :

- $V1$ est un utilitaire léger chargé de livrer et de collecter des bacs vides ;
- $V2$ est un camion toupie transportant du béton prêt à l'emploi.

Remarque 12. *Nous considérons dans ce qui suit que :*

- $V1$ dispose d'une capacité de charge suffisante pour transporter tous les bacs requis au cours d'une tournée. En d'autres termes, la contrainte de capacité de chargement est négligée pour $V1$ étant donné le poids léger des bacs vides transportés ;
- $V2$ ne peut transporter qu'une quantité limitée de béton, notée Q . Si cette quantité maximale est inférieure à la quantité requise pour satisfaire les demandes des clients au cours d'une tournée, le camion se trouvera dans l'obligation de repasser en centrale pour se ravitailler à nouveau en béton avant de reprendre sa tournée.

À $V1$ et $V2$, nous associons :

- $D1$, le dépôt de $V1$, ou, en d'autres termes, le point de départ et d'arrivée de $V1$ lors de sa tournée ;
- $D2$, le dépôt de $V2$, ou, en d'autres termes, le point de départ et d'arrivée de $V2$ lors de sa tournée.

Et nous notons $N = \{1, 2, \dots, n\}$ l'ensemble des n points devant être visités par $V1$ et/ou $V2$, N étant égal à $N_p \cup N_d$, où :

- N_d est l'ensemble des points où doivent être livrés des bacs vides, et du béton. En d'autres termes, il s'agit de l'ensemble des points nécessitant des visites successives de $V1$, puis de $V2$;
- N_p est l'ensemble des points où doit passer $V1$ pour récupérer des bacs vides déposés lors d'une précédente tournée, et dont les clients n'ont plus besoin. Le véhicule $V2$ n'est donc pas concerné par cet ensemble de points ;
- $N_d \cap N_p = \emptyset$.

Le problème peut être défini sur un graphe complet $G = (V, E)$, où :

- $V = \{D_1\} \cup \{D_2\} \cup N$ est l'ensemble des $n + 2$ points impliqués dans une tournée (les deux dépôts + les n points à visiter) ;
- $E = \{(i, j), i, j \in V, i \neq j\}$ est l'ensemble des chemins reliant les points ;
- $C = \{c_{i,j}, (i, j) \in E\}$ représente les temps de trajet séparant i et j (avec $c_{i,j} = c_{j,i}$) ;

- $D = \{d_i, i \in N\}$ indique les demandes, en nombre de bacs. $|d_i|$ représente le nombre de bacs à déposer, où à récupérer, au point i , avec :
 - $d_i < 0 \quad \forall i \in n_p$;
 - $d_i > 0 \quad \forall i \in n_d$.

Étant donné les ensembles décrits ci-dessus, et en définissant :

- $x_{i,j}$ une variable booléenne telle que :
 - $x_{i,j} = 1$ si et seulement si $V1$ (le véhicule utilitaire transportant les bacs vides) visite le point j immédiatement après avoir visité i ;
 - $x_{i,j} = 0$ sinon.
- $y_{i,j}$ une variable booléenne telle que :
 - $y_{i,j} = 1$ si et seulement si $V2$ (le camion toupie transportant le béton) visite le point j immédiatement après avoir visité i ;
 - $y_{i,j} = 0$ sinon.
- $t_{1,i}$ une variable indiquant l'instant où $V1$ quitte le point i ;
- $t_{2,i}$ une variable indiquant l'instant où $V2$ quitte le point i .

L'objectif consiste à trouver :

- un itinéraire pour le véhicule $V1$, lui permettant de visiter tous les points i tels que $i \in N = N_d \cup N_p$;
- un itinéraire pour le véhicule $V2$, lui permettant de visiter tous les points i tels que $i \in N_d$.

Cela, en minimisant le temps de trajet total, et en respectant les contraintes de précédence (tout point devant être visité par $V1$ et $V2$ doit d'abord être visité par $V1$, puis par $V2$), et de capacité de chargement ($V2$ peut transporter une quantité limitée de béton. S'il venait à l'épuiser en cours de route, il serait dans l'obligation de se ravitailler avant de poursuivre son itinéraire).

Plus formellement, notre objectif est de minimiser la somme des temps de trajet effectués par les deux véhicules :

$$\min \sum_{i=0}^n \sum_{j=0}^n x_{i,j} c_{i,j} + \sum_{i=0}^n \sum_{j=0}^n y_{i,j} c_{i,j} \quad (3.1)$$

En respectant les contraintes :

$$\sum_{j \in N} x_{i,j} = 1 \quad \forall i \in \{D_1\} \cup N \quad (3.2)$$

$$\sum_{i \in N} x_{i,j} = 1 \quad \forall j \in \{D_1\} \cup N \quad (3.3)$$

$$\sum_{j \in N_d} y_{i,j} = 1 \quad \forall i \in \{D_2\} \cup N_d \quad (3.4)$$

$$\sum_{i \in N_d} y_{i,j} = 1 \quad \forall j \in \{D_2\} \cup N_d \quad (3.5)$$

$$x_{i,D_2} = 0 \quad \forall i \in \{D_1\} \cup N \quad (3.6)$$

$$x_{D_2,i} = 0 \quad \forall i \in \{D_1\} \cup N \quad (3.7)$$

$$y_{i,j} = 0 \quad \forall i, j \in \{D_1\} \cup N_p \quad (3.8)$$

$$u_i - u_j + |N|x_{i,j} \leq |N| - 1 \quad \forall i, j \in N, i \neq j \quad (3.9)$$

$$v_i - v_j + |N_d|y_{i,j} \leq |N_d| - 1 \quad \forall i, j \in N_d, i \neq j \quad (3.10)$$

$$t_{1,j} - t_{1,i} + MAXx_{i,j} \leq MAX + c_{i,j} \quad \forall i, j \in \{D_1\} \cup N \quad (3.11)$$

$$t_{1,j} - t_{1,i} + MINx_{i,j} \geq MIN + c_{i,j} \quad \forall i, j \in \{D_1\} \cup N \quad (3.12)$$

$$t_{2,j} - t_{2,i} + MAXy_{i,j} \leq MAX + c_{i,j} \quad \forall i, j \in \{D_2\} \cup N_d \quad (3.13)$$

$$t_{2,j} - t_{2,i} + MINy_{i,j} \geq MIN + c_{i,j} \quad \forall i, j \in \{D_2\} \cup N_d \quad (3.14)$$

$$t_{1,0} = 0 \quad (3.15)$$

$$t_{2,i} \geq t_{1,i} \quad \forall i \in N_d \quad (3.16)$$

$$0 \leq u_i \leq |N| \quad \forall i \in N \quad (3.17)$$

$$0 \leq v_i \leq |N_d| \quad \forall i \in N_d \quad (3.18)$$

$$t_{1,i} \in \mathbb{N} \quad \forall i \in \{D_1\} \cup N \quad (3.19)$$

$$t_{2,i} \in \mathbb{N} \quad \forall i \in \{D_2\} \cup N_d \quad (3.20)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i, j \in \{D_1\} \cup \{D_2\} \cup N \quad (3.21)$$

- Les contraintes 3.2 et 3.3 garantissent que chaque point où doivent être livrés ou collectés des bacs soit visité exactement une fois. En d'autres termes, les contraintes garantissent que tous les points de livraison et de collecte, ainsi que le dépôt D_1 soient visités par V_1 ;
- Les contraintes 3.4 et 3.5 indiquent que tous les points de livraison, ainsi que D_2 ,

doivent être visités par $V2$;

- 3.6 et 3.7 s'assurent que $V1$ ne visite jamais $D2$ (qui est le dépôt de $V2$);
- 3.8 force $V2$ à ne jamais passer par $D1$ (le dépôt de $V1$), ni par les points faisant partie de l'ensemble N_p . En effet, cet ensemble regroupe les points de collecte, où doivent être récupérés des bacs, et qui n'ont donc pas besoin d'être approvisionnés en béton. $V2$ n'est pas concerné par ces points-là;
- 3.9 (resp. 3.10) impose que la tournée effectuée par $V1$ (resp. $V2$) soit un circuit unique, et non une union de circuits disjoints.
- Les contraintes 3.11 et 3.12 concernent la définition de $t_{1,i}$ (l'instant où $V1$ quitte le point i). Ainsi :
 - si $V1$ quitte un point i pour se diriger immédiatement après vers un point j (en d'autres termes, lorsque $x_{i,j} = 1$), alors d'après 3.11 et 3.12, on aurait :

$$t_{1,j} - t_{1,i} + MAX \leq MAX + c_{i,j}$$

$$t_{1,j} - t_{1,i} + MIN \geq MIN + c_{i,j}$$

Ce qui implique :

$$t_{1,j} - t_{1,i} \leq c_{i,j}$$

$$t_{1,j} - t_{1,i} \geq c_{i,j}$$

Et donc :

$$t_{1,j} = t_{1,i} + c_{i,j}$$

En d'autres termes, d'après nos hypothèses, l'instant où un véhicule quitte un point j se calcule en additionnant l'instant où il quitte i , le point précédent dans la tournée, et le temps de trajet nécessaire entre i et j . Notons donc que les temps d'immobilisation des véhicules à chaque point sont négligés.

- si $V2$ ne visite pas j immédiatement après i , alors $x_{i,j} = 0$, et donc, d'après 3.11 et 3.12, on aurait :

$$t_{1,j} - t_{1,i} \leq MAX + c_{i,j}$$

$$t_{1,j} - t_{1,i} \geq MIN + c_{i,j}$$

Sachant que MAX (resp. MIN) est un entier suffisamment grand (resp. petit), les contraintes 3.11 et 3.12 sont toujours satisfaites pour $x_{i,j} = 0$

- Les contraintes 3.13 et 3.14 définissent $t_{2,i}$ (l'instant où $V2$ quitte chaque point i qu'il visite) de la même manière que les contraintes 3.11 et 3.12, présentées ci-dessus, définissent $t_{1,i}$;
- 3.15 indique que $V1$ quitte son dépôt $D1$ à $t_{1,0} = 0$. La sortie de $V1$ de son dépôt pour entamer sa tournée marque donc l'instant 0 de notre repéré temporel ;
- La contrainte 3.16 implique que $V1$ doit toujours précéder $V2$ à chaque point où la visite des deux véhicules est requise.

3.5 Contribution

Après avoir décrit le schéma logistique considéré (voir section 3.1), et après avoir formellement défini le problème sous-jacent (section 3.4), nous exposons ci-après les approches que nous avons proposées et testées pour répondre à la problématique posée.

La première approche se déroule en deux étapes :

- d'abord, une tournée de livraison et de collecte de bacs est générée pour le véhicule $V1$ à l'aide d'une recherche locale. Le véhicule $V2$, ainsi que la tournée associée, ne sont à ce stade pas pris en compte ;
- ensuite, une fois la tournée de $V1$ fixée, la seconde tournée, celle de $V2$, est construite en considérant l'itinéraire préétabli de $V1$ comme une contrainte forte.

Cette approche est dite **séquentielle** car les deux tournées sont calculées l'une après l'autre. La première tournée générée contraint fortement les choix qui peuvent être faits lors de la construction de la tournée suivante.

Le seconde approche proposée consiste quant à elle à générer simultanément les deux tournées, l'une pouvant s'adapter à l'autre au besoin. Pour ce faire, un algorithme basé sur un recuit simulé recherche la meilleure "combinaison de tournées" possible. Les deux itinéraires sont donc construits en même temps, avec plus de flexibilité. Pour cette approche, on parlera d'optimisation **simultanée**.

3.6 Optimisation séquentielle

Comme cela a été évoqué plus haut, la stratégie d'optimisation séquentielle se déroule en deux temps :

1. calculer la tournée de livraison et de collecte de bacs qui servira d'itinéraire au véhicule $V1$;
2. en partant de la tournée de $V1$, calculer la tournée de livraison de béton, qui représentera l'itinéraire de $V2$.

3.6.1 Construction de la tournée de livraison et de collecte de bacs

L'algorithme utilisé pour générer une tournée pour $V1$ se base sur une recherche locale similaire à celle présentée dans la section 2.7. On part d'une solution initiale S , et on tente de l'améliorer, au fur et à mesure, en explorant des solutions de son voisinage. Comme expliqué dans la section 2.7, la procédure est relancée à plusieurs reprises avec différentes solutions initiales (générées à l'aide de diverses heuristiques) afin de minimiser les risques de tomber sur des optimum locaux.

A chaque itération d'une exécution de l'algorithme, le voisinage *1-Shift* [Bertsimas, 1988] de la solution courante est généré et exploré à la recherche d'une "meilleure" solution, qui minimise le temps de trajet total de la tournée de livraison et de collecte. L'algorithme s'arrête lorsqu'aucune amélioration n'est possible.

Il est à noter que toutes les solutions explorées par l'algorithme sont cohérentes du moment qu'elles prennent en compte tous les points devant être visités. En effet, la contrainte de capacité de chargement étant négligée pour le véhicule $V1$, il n'y a pas besoin d'effectuer de vérifications pour filtrer les solutions violant cette contrainte.

3.6.2 Construction de la tournée de livraison de béton

Une fois que l'itinéraire prévu pour $V1$ est fixé, celui de $V2$ est construit en prenant en compte la tournée de $V1$ comme contrainte forte.

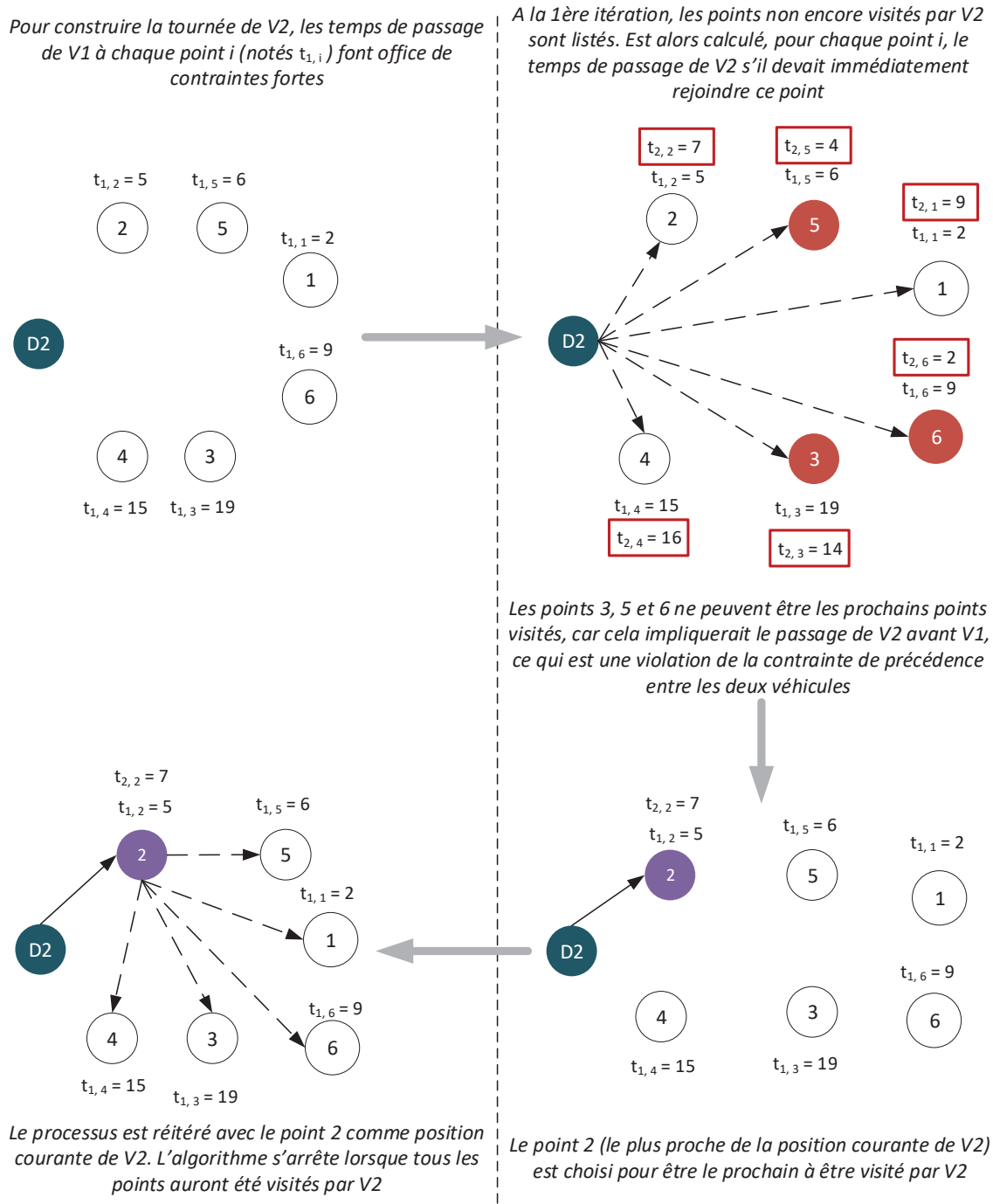


FIGURE 3.3 – Construction d'une tournée de livraison de béton

Comme présenté dans la figure 3.3, l'itinéraire de V2 est construit au fur et à mesure, itération après itération. Ainsi, lorsque V2 s'apprête à quitter un point i :

1. tous les points de livraison non encore visités sont listés ;
2. la valeur $t_{2,i} + c_{i,j} + t_r$ est calculée pour chacun des points j appartenant à l'ensemble des points à visiter. En d'autres termes, on cherche, pour chaque point j devant être visité, l'instant où V2 atteindrait ce point s'il était choisi comme prochaine

étape dans la tournée. t_r représente ici le temps de trajet nécessaire au véhicule pour se réapprovisionner s'il ne transporte pas la quantité suffisante de produit pour satisfaire la demande de j . t_r est donc égal à :

- 0 si $V2$ possède une quantité suffisante de produit pour satisfaire la demande de j (aucun détour pour réapprovisionnement n'est nécessaire) ;
- $c_{i,C} + c_{C,j}$, si un réapprovisionnement en centrale est nécessaire. Il s'agit de la somme des temps de trajet entre :
 - i , l'emplacement actuel de $V2$ et C , la centrale à béton dans laquelle il doit se réapprovisionner, et
 - la centrale à béton et j , le prochain point à visiter.

3. on élimine tous les points j pour lesquels $t_{2,i} + c_{i,j} + t_r < t_{1,j}$, $t_{1,j}$ étant l'instant où $V1$ arrive au point j (cette information est connue étant donné que la tournée de $V1$ a déjà été fixée). Dit autrement, on élimine tous les points qui, s'ils devaient être les prochains à être visités par $V2$, provoqueraient une violation de la contrainte de précédence qui impose que, en tout point, la visite de $V1$ soit impérativement antérieure à celle de $V2$;
4. parmi tous les points non encore visités, et respectant la contrainte de précédence, $V2$ se dirige vers le plus proche en termes de temps de trajet (celui qui minimise $t_{2,i} + c_{i,j} + t_r$).

3.7 Optimisation simultanée

La stratégie d'optimisation séquentielle présentée plus haut traite les tournées des véhicules $V1$ et $V2$ séparément. L'approche simultanée exposée ci-dessous s'appuie sur un recuit simulé, et vise à donner plus de flexibilité lors de la construction des deux tournées. Elle calcule les deux itinéraires simultanément pour que chacun puisse s'adapter à l'autre.

3.7.1 Principe

Le recuit simulé est une métaheuristique proposée par [Kirkpatrick et al., 1983]. Elle s'inspire d'un processus de refroidissement lent et de réchauffage utilisé en métallurgie. Le recuit simulé est omniprésent dans la littérature, où il permet souvent d'obtenir de résultats très satisfaisants pour diverses typologies de problèmes d'optimisation, y compris

les problèmes de tournées de véhicules [Chiang and Russell, 1996, Czech and Czarnas, 2002, Van Breedam, 1995]...

Comme indiqué dans l’algorithme 4, un recuit simulé part d’une solution courante S , et essaye de sélectionner, à chaque itération, une solution voisine candidate S' , avec une certaine probabilité $p = e^{-\Delta/T}$. Cette probabilité change à chaque itération selon les variations des deux paramètres qui la composent :

- la température T . Il s’agit d’un nombre strictement positif qui prend sa valeur la plus haute T_0 à la première itération de l’algorithme, et dont la valeur décroît progressivement itération après itération. Notons que plus élevée est la valeur de T , plus grande est la probabilité qu’une solution candidate soit acceptée comme prochaine solution courante ;
- $\Delta = f(S') - f(S)$. Il s’agit de la différence entre les valeurs de la fonction objectif f appliquée, respectivement à la solution courante S et à sa solution voisine candidate pour la remplacer S' . Pour rappel, la fonction objectif considérée ici est le temps de trajet total effectué par les deux véhicules $V1$ et $V2$ durant leurs tournées respectives. Il s’agit donc de minimiser f .

Ainsi donc, si $f(S') - f(S) < 0$, alors cela signifie que S' est une meilleure solution que S . Dans ce cas-là, $\Delta < 0 \implies e^{-\Delta/T} > 1$ (car $T > 0$). Et donc, S' est automatiquement choisie pour prendre la place de S .

Par contre, si $f(S') - f(S) > 0$, alors S' est choisie avec une certaine probabilité $e^{-\Delta/T} < 1$. Plus importante est la différence entre $f(S')$ et $f(S)$, plus réduite est la probabilité que S' soit acceptée comme prochaine solution courante. Le recuit simulé ne s’interdit donc pas d’accepter une solution de moins bonne qualité que la solution courante, mais il accepte de moins en moins facilement des solutions de moins en moins bonnes.

3.7.2 Représentation d’une solution

D’un point de vue algorithmique, une solution manipulée par le recuit simulé n’est rien d’autre qu’un couple de vecteurs unidimensionnels, faisant respectivement référence à la tournée devant être effectuée par le véhicule $V1$ et à celle associée à $V2$. Dans chacun des vecteurs, l’élément se trouvant à la $i^{\text{ème}}$ position représente le $i^{\text{ème}}$ point visité par le véhicule associé au vecteur en question (voir figure 3.4).

Algorithme 4 : Principe du recuit simulé

```

iterationsWithoutImprovement ← 0 ;
/* La température prend sa valeur initiale  $T_0$  */
T ←  $T_0$  ;
/* La solution courante initiale est générée à l'aide d'une
   heuristique */
S ← initialSolution() ;
bestSolution ← S ;
N ← ∅ ;
/* Tant que le nombre d'itérations maximal sans amélioration n'est
   pas atteint... */
while (iterationsWithoutImprovement < maxIterations) do
    iterationsWithoutImprovement ++ ;
    /* Le voisinage de la solution courante est exploré */
    N ← Neighborhood(S) ;
    /* La meilleure solution voisine non encore sélectionnée est
       choisie pour être la solution candidate */
    S' ← BestNeighbour(N) ;
    Δ ← f(S') - f(S) ;
    p ← Random(0,1) ;
    /* La solution candidate est choisie pour remplacer la solution
       courante avec une probabilité p */
    if (p ≤ e-Δ/T) then
        S ← S' ;
        if (f(S) ≤ f(bestSolution)) then
            bestSolution ← S ;
            iterationsWithoutImprovement ← 0 ;
    /* La température T est mise à jour. Elle est multipliée par un
       coefficient de décroissance α ∈ ]0,1[ */
    T ← αT ;

```

Remarque 13. La taille du vecteur lié au véhicule $V1$ est égal à $|n_p| + |n_d| + 1$. En d'autres termes, il s'agit du nombre de points devant être visité par $V1$ (les points de livraison + les points de collecte + le dépôt de $V1$).

Pour le vecteur lié à $V2$, sa taille est de $|n_d| + 1$ (les points de livraison + le dépôt de $V2$).

3.7.3 Définition du voisinage

Le recuit simulé est une méthode qui s'appuie sur l'exploration, itération après itération, du voisinage d'une solution courante. C'est pourquoi il est nécessaire de définir ce qu'est une solution voisine.

Pour générer une solution voisine d'une solution donnée S , notre algorithme commence

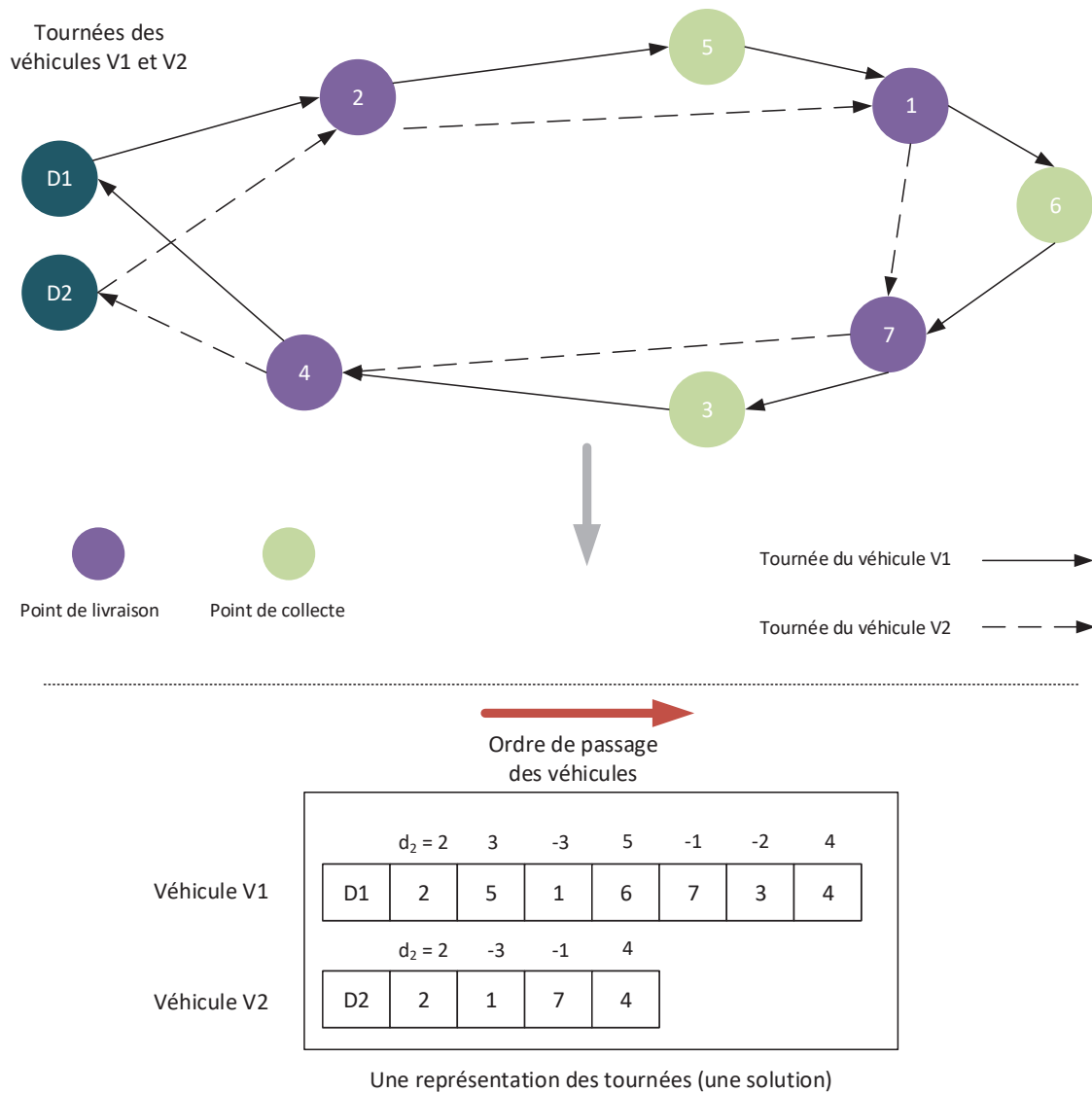


FIGURE 3.4 – Représentation des tournées

par choisir, aléatoirement, deux différents points i et j parmi tous les points à visiter. Ce peuvent être deux points de livraison, deux points de collecte, ou un point de livraison et un point de collecte. On distingue alors deux cas :

1. si au moins un des points sélectionnés est un point de collecte (pour rappel, les points de collecte ne sont visités que par le véhicule $V1$), alors la solution voisine sera générée en prenant la solution de départ S , et en intervertissant les positions des points i et j dans la tournée du véhicule $V1$. Par exemple, si dans la solution de départ S , le point i est le premier visité, et j le troisième visité par $V1$, alors, dans la solution voisine, i sera le troisième point, et j le premier point visité par $V1$. L'ordre de passage reste inchangé pour les autres points visités par $V1$, et la

tournée de $V2$ ne subit aucune modification ;

2. si les deux points choisis aléatoirement sont tous les deux des points de livraison, alors, les positions des points i et j sont intervertis, à probabilités égales :
 - dans la tournée de $V1$, ou
 - dans la tournée de $V2$, ou
 - dans les deux tournées simultanément (voir l'exemple de la figure 3.5).

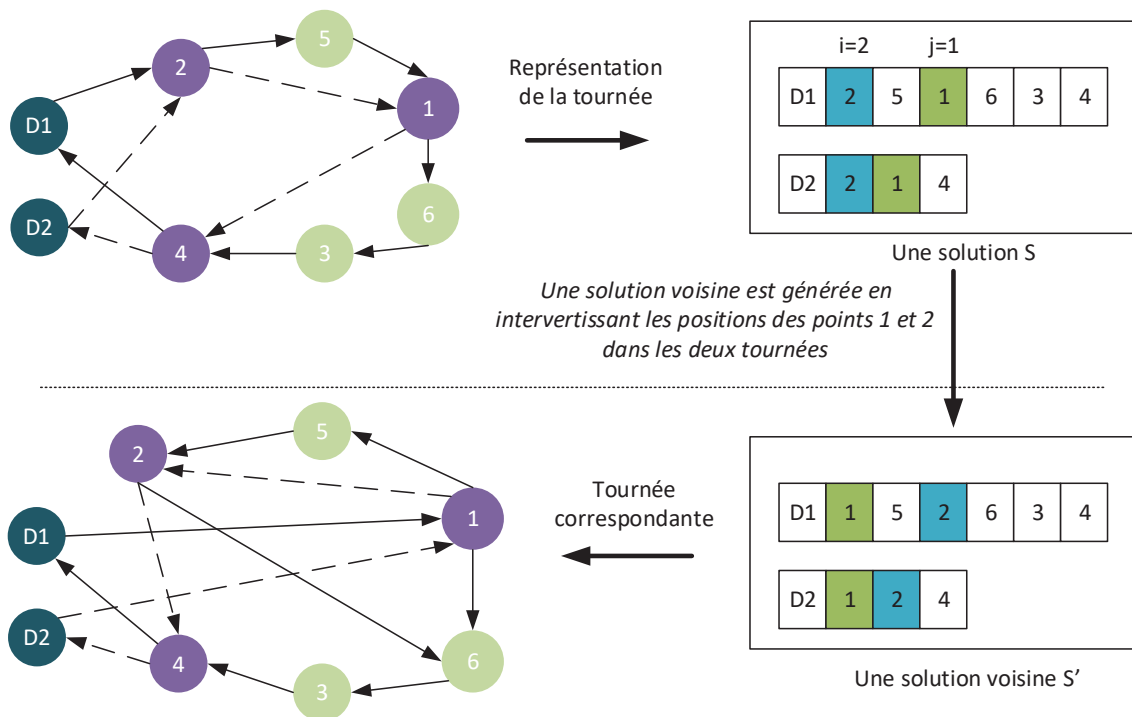


FIGURE 3.5 – Génération d'une solution voisine

Il est à noter qu'à chaque fois qu'une solution est générée, on part du principe que $V1$ quitte son dépôt à l'instant $t_{1,0} = 0$, et que $V2$ entame sa tournée à un instant $t_{2,0} = T$ où T est l'instant le plus proche possible de $t = 0$ et qui permet de garantir la contrainte de précédence entre $V1$ et $V2$. En d'autres termes, $V2$ quitte son dépôt le plus tôt possible, du moment que $V1$ a suffisamment d'avance pour ne jamais passer après $V2$ pour un des nœuds qu'ils doivent visiter tous les deux. De cette manière les tournées générées sont toujours faisables au regard de la contrainte de précédence.

Remarque 14. *Il est à noter que le temps T passé par $V2$ au dépôt, en attendant que $V1$ prenne suffisamment d'avance, est inclus dans la durée de trajet totale de $V2$.*

3.8 Expérimentations

Nous présentons ici les tests auxquels les deux approches (optimisation séquentielle et optimisation simultanée) ont été soumises.

3.8.1 Implémentation

Les algorithmes précédemment présentés ont été développés :

- en JAVA ;
- Sous LINUX (UBUNTU 18.04) ;
- sur une machine dotée d'un processeur AMD A10-7700K RADEON R7, 3.40 GHz et de 8 GB de RAM.

De plus, le programme linéaire en nombre entiers présenté dans la section 3.4 a été implémenté sur CPLEX.

3.8.2 Jeu de données

Les expérimentations ont été menées sur des instances standards, largement utilisées dans la littérature. Ces instances ont été générées par [Gendreau et al., 1999] pour le PDTSP (*Pickup & Delivery Travelling Salesman problem*). Chaque instance est représentée par :

- une matrice où l'élément (i, j) correspond au coût du trajet entre les points i et j , cela correspond aux valeurs $c_{i,j}$ présentées précédemment. Les éléments de la matrice respectent les propriétés mathématiques définissant une distance, à savoir, la symétrie, la séparation, et l'inégalité triangulaire ;
- un vecteur dont l'élément i correspondant à la demande du point i . Dans notre cas, il s'agira du nombre de bacs à livrer, ou à récupérer.

Les instances sont scindées en 6 classes selon le nombre de points à visiter : 25 points, 50 points, 75 points, 100 points, 150 points et 200 points. Chacune des classes est composée de 30 instances, ce qui fait en tout 180 instances.

Cela étant dit, nous avons dû effectuer quelques modifications au jeu de données originel afin qu'il corresponde aux besoins de notre problématique :

- pour chacune des instances considérées, nous avons enlevé un point à visiter (le premier), que nous considérons comme le dépôt (ou point de départ) du véhicule

$V2$ (le dépôt initialement prévu dans l'instance est attribué au véhicule $V1$) ;

- le dépôt attribué à $V2$ est également considéré comme le point de réapprovisionnement du véhicule $V2$ s'il n'était plus capable de satisfaire les demandes des clients en cours de route.

La taille des voisinages générés par nos algorithmes est fixée à 200. Ce nombre offre un compromis intéressant entre qualité des solutions obtenues et vitesse d'exécution des algorithmes.

Pour finir, et pour ce qui est de l'approche basée sur le recuit simulé, les paramètres T_0 et α , représentant respectivement la température initiale du recuit, et le coefficient de décroissance de la température ont été fixés à 4800 et 0.95. Plusieurs combinaisons ont été testées, et c'est avec ces valeurs que les meilleurs résultats ont été obtenus.

3.8.3 Résultats

Le tableau 3.1 montre le coût moyen des solutions obtenues avec la stratégie d'optimisation séquentielle (voir section 3.6) pour chacune des classes d'instances de [Gendreau et al., 1999].

La première colonne désigne le nombre de nœuds de la classe d'instances considérée, la deuxième représente le coût moyen total des tournées générées. Les troisième et quatrième colonnes détaillent respectivement les parts du premier et du second véhicule dans le coût total des tournées. La cinquième colonne représente quant à elle le temps d'exécution moyen de l'algorithme en millisecondes.

La première remarque que l'on peut formuler à partir des résultats du tableau est que,

Instances	Total des deux tournées	Tournée du véhicule $V1$	Tournée du véhicule $V2$	Temps CPU moyen (ms)
25	929	565 (60.8%)	364 (39.2%)	83
50	1445	872 (60.3%)	573 (39.7%)	399
75	1924	1144 (59.4%)	780 (40.6%)	1001
100	2432	1430 (58.8%)	1002 (41.2%)	1878
150	3634	2144 (59%)	1490 (41%)	5897
200	4519	2705 (59.8%)	1814 (40.2%)	15532

TABLE 3.1 – Coût moyen (ou durées de trajet moyennes) des solutions obtenues avec la stratégie d'optimisation séquentielle sur les instances de [Gendreau et al., 1999]

dans le coût global des tournées obtenues, la part des tournées effectuées par $V1$ (celles qui consistent à livrer et collecter des bacs) est plus importante que celle des tournées

du véhicule $V2$. En effet, les coûts moyens engendrés par le parcours de $V1$ représentent environ 60% du coût moyen total. Cela s'explique par le fait que $V1$ doit effectuer un plus grand nombre de visites que $V2$. Pour rappel, $V1$ se charge des livraisons et des collectes, il doit donc visiter tous les clients requérant des livraisons et tout ceux demandant des collectes. Le véhicule $V2$ quant à lui ne s'occupe que du sous-ensemble de clients requérant des livraisons.

Néanmoins, lorsqu'on sait que, pour les instances considérées, l'ensemble des nœuds requérant des livraisons est équivalent à celui des nœuds requérant des collectes (en termes de cardinalité), on tend à penser que, finalement, les coûts engendrés par $V1$ ne sont pas aussi importants que ce qu'on aurait pu prévoir. En effet, $V1$ visite deux fois plus de nœuds que $V2$, mais n'engendre pas plus de 60% du coût total des tournées.

Ce phénomène peut s'expliquer par le comportement de l'algorithme à optimisation séquentielle qui commence par optimiser la tournée de $V1$ avant de construire une tournée pour $V2$. Une telle façon de faire donne un avantage à la tournée construite en premier car elle n'est pas contrainte par la seconde. l'itinéraire prévu pour $V1$ est mieux optimisé que celui de $V2$, d'où l'impact sur le coût total des tournées qui est moins important que prévu pour $V1$.

Le tableau 3.2 montre le coût moyen de solutions obtenues en soumettant les instances de [Gendreau et al., 1999] à la stratégie d'optimisation simultanée (voir section 3.7). Là encore, la première colonne représente la classe d'instances, la deuxième montre le coût moyen total des tournées. Les colonnes 3 et 4 donnent le détail de la répartition des coûts sur les deux tournées. Comme pour la stratégie d'optimisation séquentielle, la part des

Instances	Total des deux tournées	Tournée du véhicule V1	Tournée du véhicule V2	Temps CPU moyen (ms)
25	908	530 (58%)	377(42%)	95
50	1266	685 (54%)	580 (46%)	457
75	1467	837(57%)	629 (43%)	1046
100	1692	967 (57%)	724 (43%)	2029
150	2047	1216 (57%)	830 (43%)	6719
200	2451	1462 (57%)	988 (43%)	18198

TABLE 3.2 – Coût moyen (ou durées de trajet moyennes) des solutions obtenues avec la stratégie d'optimisation simultanée sur les instances de [Gendreau et al., 1999]

tournées du véhicule $V1$ dans les durées de trajet totales est supérieure à celle des tournées

de $V2$. Aussi, de la même manière que pour la stratégie séquentielle, la part de $V1$ est moins importante que ce qu'on pourrait attendre étant donné que ce véhicule visite deux fois plus de nœuds que $V2$. En effet, les durées de trajet moyennes de $V1$ représentent environ 57% des durées de trajet totales (ce chiffre est d'ailleurs un peu plus bas qu'avec la stratégie séquentielle où il tourne autour des 60%).

Là encore, les résultats laissent à penser que les tournées de $V1$ sont mieux optimisées que celles de $V2$. Pour la stratégie séquentielle, le fait de construire l'itinéraire de $V1$ avant celui de $V2$ a certainement pu donner un avantage au premier véhicule, mais avec la stratégie simultanée, les deux parcours sont générés en même temps.

L'explication ici est sans doute liée à la contrainte qui impose que $V1$ précède toujours $V2$ à chaque nœud devant être visité par les deux. En effet, les tournées du second véhicule sont contraintes par cette condition. $V2$ ne peut quitter son dépôt que lorsque $V1$ a pris suffisamment d'avance pour que la contrainte de précédence ne soit jamais violée, et le temps passé par $V2$ à attendre, immobile dans son dépôt, est autant de temps perdu (et comptabilisé dans les coûts de fonctionnement des tournées). $V1$ quant à lui ne s'immobilise jamais car il n'a pas à se soucier d'un autre véhicule qui devrait passer avant lui. Les durées de trajet effectuées par $V1$ sont donc "productives" à 100% contrairement à celles de $V2$ qui peuvent inclure des périodes d'immobilisation qui ont un coût et qui ne rapportent rien.

Par ailleurs, une autre raison qui explique également, en partie, la répartition constatée des coûts entre les deux véhicules est le fait que $V2$, contrairement à $V1$ est amené à retourner au dépôt pendant les tournées s'il y a besoin de se réapprovisionner avant de poursuivre son parcours. $V2$ visite donc moins de nœuds que $V1$, mais il est sujet à des détours qui rallongent son temps de parcours total.

Le tableau 3.3 compare les approches simultanée et séquentielle en termes de coût global, et de coût par véhicule.

Il apparait clairement que la stratégie simultanée présente de meilleures performances que la séquentielle. Le fait de générer les tournées des deux véhicules en même temps semble permettre une meilleure optimisation globale.

On remarque également que l'écart entre les deux approches se creuse à mesure que la taille des instances considérées augmente. Ainsi, on passe d'un écart moyen de 2% pour les

Instances	Total des deux tournées		V1		V2	
	Optimisation simultanée	Optimisation séquentielle	Optimisation simultanée	Optimisation séquentielle	Optimisation simultanée	Optimisation séquentielle
25	908	929 (+2.3%)	530	565 (+6.6%)	377	364 (-3.4%)
50	1266	1445 (+14.3%)	685	872 (+27.2%)	580	573 (-1.2%)
75	1467	1924 (+31.1%)	837	1144 (+36.6%)	629	780 (+24%)
100	1692	2432 (+43.7%)	967	1430 (+47.8%)	724	1002 (+38.3)
150	2047	3634 (+77.5%)	1216	2144 (+76.3%)	830	1490 (+79.5%)
200	2451	4519 (+84.3%)	1462	2705 (+85%)	988	1814 (+83.6%)

TABLE 3.3 – Optimisation simultanée VS Optimisation séquentielle

plus petites instances à 84% pour les plus grandes. Le constat est le même en prenant les tournées de $V1$ et $V2$ séparément. Le recours à la stratégie simultanée est donc d'autant plus intéressant pour des problèmes de grande taille, où l'espace de recherche est plus vaste, et où le besoin de flexibilité et d'adaptation continue des tournées l'une à l'autre est plus fort.

Afin d'avoir un repère concernant la qualité des solutions obtenues par nos approches, le programme linéaire en nombres entiers (PLNE) décrit dans la section 3.4 a été implémenté et testé lui aussi sur les instances de [Gendreau et al., 1999].

Le tableau 3.4 compare ainsi les résultats optimaux ainsi obtenus à ceux trouvés avec la stratégie d'optimisation simultanée, celle-ci s'étant montrée plus performante que la stratégie séquentielle.

Malheureusement, le PLNE n'a pu trouver de solutions que pour les classes ayant les

Instances	PLNE			Optimisation simultanée		
	Total des deux tournées	V1	V2	Total des deux tournées	V1	V2
25	803	464	339	908 (+13%)	530 (+14.2%)	377 (+11.2%)
50	1059	587	472	1266 (+19.5%)	685 (+16.7%)	580 (+22.8%)

TABLE 3.4 – Solutions optimales VS Optimisation simultanée

instances les plus petites (25 et 50 nœuds). Pour les autres, aucune solution n'a pu être

trouvée en un temps raisonnable. Il est donc difficile d'analyser précisément les résultats dans ces conditions, et il est également difficile de se comparer à d'autres travaux étant donné le déroulement spécifiques des tournées étudiées ici, et dont nous n'avons pas trouvé de cas comparables dans la littérature. On voit néanmoins que la stratégie d'optimisation simultanée obtient des résultats satisfaisants, les écarts par rapport aux solutions optimales étant inférieurs à 20% pour des instances allant jusqu'à 50 nœuds.

Conclusion

Dans ce chapitre, nous avons présenté le second schéma logistique proposé en vue d'optimiser le processus de livraison de béton en petites quantités.

Le scénario étudié fait appel à deux véhicules. Le premier, un utilitaire léger, livre des bacs vides conçus pour contenir du béton frais chez un certain nombre de clients qui en ont fait la demande tout en collectant des bacs vides ayant été livrés lors de précédentes tournées, et dont les clients n'ont plus besoin.

En parallèle à ce travail de livraisons et de collectes de bacs, un second véhicule, un camion toupie, transporte du béton frais et effectue lui aussi une tournée en ne visitant, pour sa part, que les clients devant se faire livrer des bacs par le premier véhicule. Ainsi, pour résumer, le processus consiste, d'abord à livrer des bacs vides à l'aide d'un premier véhicule, puis à remplir les bacs avec du béton livré dans la foulée par un second véhicule. Le problème posé par ce schéma logistique est un problème de tournées de véhicules avec contraintes de synchronisation. En effet, certains clients doivent être visités par les deux véhicules, et dans un ordre prédéfini (les bacs doivent toujours être livrés avant le béton). Il s'agit d'un sujet assez peu étudié dans la littérature par rapport à des tournées plus classiques.

Après avoir formalisé le problème et clairement défini ses contraintes, nous avons proposé deux stratégies différentes pour y répondre. La première, la stratégie d'optimisation séquentielle, consiste à générer une tournée de collectes et de livraisons de bacs en premier, puis de considérer cette tournée comme contrainte forte pour construire le parcours du second véhicule. La deuxième stratégie quant à elle, la stratégie d'optimisation simultanée, permet de construire les deux tournées en même temps, les parcours des deux véhicules s'adaptant l'un à l'autre pour trouver le meilleur équilibre possible.

Les deux approches ont été testées sur des datasets standards, largement utilisés dans la littérature, et les résultats obtenus sont satisfaisants, notamment pour ce qui est de l'optimisation simultanée.

Conclusion générale et perspectives

Le travail présenté dans ce document est le fruit d'une collaboration entre le Laboratoire d'Informatique en Images et Systèmes d'Information (LIRIS) et BÉTON DIRECT, une startup Lyonnaise spécialisée dans la vente de béton frais aux petits consommateurs. À sa création, l'entreprise s'est donné pour mission de digitaliser un secteur, le BTP, où l'outil numérique est bien trop peu exploité, et ce afin de proposer la meilleure expérience d'achat possible à sa clientèle, souvent négligée par les grands acteurs du secteur.

Dans sa démarche de simplification du parcours d'achat de béton frais pour ses clientèle cible, qui est constituée de petits consommateurs de béton, l'entreprise a identifié un certain nombre de pistes à explorer, parmi lesquelles on trouve l'adaptation de la logistique de livraison de béton aux petites quantités. En effet, la livraison classique de béton frais passe par la mobilisation de camions lourds, coûteux, et polluants pour chaque commande effectuée par un client, quelle que soit la quantité commandée.

Aussi, le point de départ qui a donné naissance à cette thèse a été le besoin pour l'entreprise de développer un nouveau mode de livraison, mieux adapté aux petites quantités, moins cher pour les clients, et moins émetteur de CO₂. Très vite, l'idée du recours à des tournées de véhicules afin de mutualiser les matériels de transport a émergé. En effet, exploiter un véhicule pour effectuer des tournées durant lesquelles plusieurs petits consommateurs recevraient leur béton frais dans des bacs spécialement conçus pour contenir ce matériau semble beaucoup plus avantageux que de mobiliser un véhicule à chaque fois qu'une commande est effectuée. Néanmoins, une telle solution demande, entre autres, le développement d'un outil informatique capable de construire et de gérer les tournées de manière efficace, en respectant toutes les contraintes du métier sous-jacentes.

Deux schémas de livraison ont été proposés et étudiés durant la thèse.

Le premier schéma, détaillé dans le chapitre 2, implique un unique véhicule qui doit, pendant sa tournée, livrer des bacs remplis de béton à un certain nombre de clients, tout en collectant des bacs vides chez des clients livrés lors de précédentes tournées, le tout en minimisant les coûts associés. Ce type de tournées, qui a déjà fait l'objet de travaux dans la littérature, s'apparente à ce qui est communément appelé "tournée du laitier". Par ailleurs, pour ce premier schéma logistique, une contrainte supplémentaire a été étudiée. Il s'agit d'acheminer, en cours de tournée, d'éventuels résidus de béton non consommé récupérés chez certains clients. Ainsi, à chaque fois que du béton non consommé est collecté, le véhicule doit se détourner de son itinéraire originel pour se diriger vers un centre de recyclage, où le béton non consommé est déposé, avant de poursuivre sa tournée. Cette contrainte introduit les notions de dynamisme et d'incertitude dans les tournées, car le fait de devoir passer en centre de recyclage à tel ou tel moment, et la fréquence de ces passages, sont autant d'informations dont on ne dispose pas l'avance, et qui ne sont révélées que pendant le parcours du véhicule. Dans la littérature, on parle de tournées de véhicules dynamiques et/ou stochastiques.

Pour ce premier problème, deux approches de résolution ont été développées et testées sur des instances standards. La première, l'approche "stochastique", se base sur des probabilités de passage en centres de recyclage pour anticiper les éventuels détours devant être effectués en cours de route, et faire ainsi en sorte que leur impact soit le plus réduit possible sur la qualité des tournées générées. La seconde approche quant à elle, l'approche "dynamique", est capable de mettre à jour en cours de route les tournées préalablement générées, en prenant en compte les informations révélées au fil de l'eau.

Ce premier schéma logistique présente l'avantage d'être assez simple à déployer d'un point de vue logistique car il ne requiert qu'un seul véhicule qui s'occupe de livrer des bacs pleins et de les collecter vides. Cependant, le fait de devoir livrer des bacs remplis de béton limite la quantité de produit transportable durant une tournée (et donc le nombre de clients pouvant être servis), et ce à cause du volume occupé par les bacs remplis, qui ne peuvent donc pas être empilés et qui saturent vite la capacité de chargement du véhicule.

Pour cette raison, un second schéma logistique a été étudié, et il fait cette fois-ci appel à deux véhicules (voir chapitre 3). Le principe est d'avoir un premier véhicule qui effectue des livraisons et des collectes de bacs vides, tandis qu'un second, un camion toupie, se charge de transporter le béton. L'idée est de faire livrer, par un premier véhicule, et chez

chaque client ayant commandé du béton, des bacs vides conçus pour contenir du béton. Puis, de faire passer un second véhicule chez ces mêmes clients, pour remplir les bacs précédemment reçus.

Il s'agit ici d'un problème de tournées de véhicules avec contraintes de synchronisation et de précedence dans la mesure où deux véhicules doivent se coordonner pour visiter un ensemble de clients dans un ordre prédéfini. Là encore, deux stratégies ont été développées et testées pour résoudre le problème. Dans la première, la stratégie d'optimisation "séquentielle", il est question de générer la tournée du premier véhicule, puis de construire l'itinéraire du second véhicule en l'adaptant au premier. La seconde stratégie, la stratégie d'optimisation "simultanée", génère quant à elle les itinéraires des deux véhicules en même temps, les deux s'adaptant l'un à l'autre.

La figure 3.6 résume les différents schémas et approches étudiés.

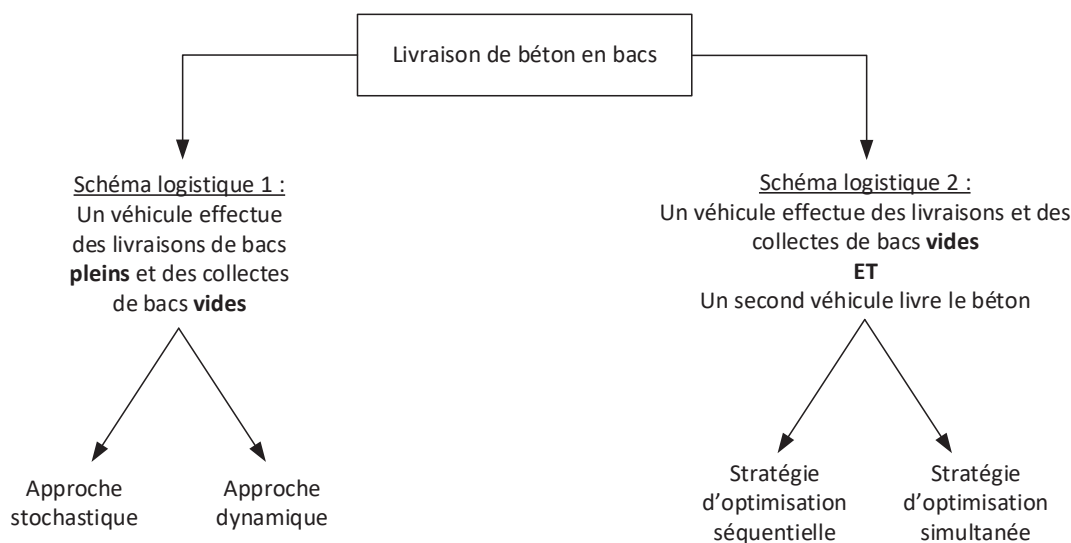


FIGURE 3.6 – Résumé des schémas et approches étudiés

De nombreuses perspectives peuvent être envisagées au travail exposé dans ce document. Parmi les pistes qu'il serait intéressant de creuser, on peut par exemple :

- explorer d'autres méthodes de résolution, exactes, mais aussi approchées. Les méta-heuristiques à base de populations (colonies de fourmis, algorithmes génétiques...) en sont un exemple ;
- tester le parallélisme, notamment sur GPU. À titre d'exemple, il serait intéressant d'observer le comportement d'une approche par échantillonnage, similaire à celle présentée dans la section 2.7, sur une architecture hautement parallèle de type

CUDA. En effet, générer un grand nombre de scénarios de livraison possibles et les évaluer est un travail qui se prête très bien au type de parallélisme offert par les GPU. Cela pourrait permettre de générer beaucoup plus de scénarios, beaucoup plus rapidement ;

- intégrer de nouvelles contraintes dans les deux schémas étudiés. On pourrait envisager, entre autres, l'ajout de fenêtres de temps durant lesquelles les clients peuvent être visités, ou encore considérer des tournées impliquant un plus grand nombre de véhicules ;
- inclure d'autres objectifs dans le processus d'optimisation. Pour le second schéma par exemple, il serait pertinent d'intégrer la durée d'attente entre les passages du premier et du second véhicule comme variable à optimiser car il s'agit d'un paramètre important d'un point de vue "expérience client"...

Liste des publications

- Hadjadj, S. & Kheddouci, H. (2018). « Estimation-based algorithm for a stochastic one-commodity pick-up & delivery travelling salesman problem ». 7th International Conference on Metaheuristics and Nature Inspired Computing, 27 octobre 2018, Marrakech (Maroc), (pp. 331-338).
- Hadjadj, S. & Kheddouci, H. (2019). « A heuristic algorithm for a vehicle routing problem with pickup & delivery and synchronization constraints ». 9th International Network Optimization Conference (INOC), 12 juin 2019, Avignon (France), (pp. 113-117).
- Hadjadj, S., & Kheddouci, H. (2020). Simulated Annealing Approach for the Vehicle Routing Problem with Synchronized Visits. In ICORES (pp. 297-303).

Bibliographie

- [Afifi et al., 2016] Afifi, S., Dang, D.-C., and Moukrim, A. (2016). Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. *Optimization Letters*, 10(3) :511–525.
- [Alshamrani et al., 2007] Alshamrani, A., Mathur, K., and Ballou, R. H. (2007). Reverse logistics : simultaneous design of delivery routes and returns strategies. *Computers & Operations Research*, 34(2) :595–619.
- [Angelelli and Mansini, 2002] Angelelli, E. and Mansini, R. (2002). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In *Quantitative approaches to distribution logistics and supply chain management*, pages 249–267. Springer.
- [Anily et al., 1999] Anily, S., Gendreau, M., and Laporte, G. (1999). The swapping problem on a line. *SIAM Journal on Computing*, 29(1) :327–335.
- [Anily et al., 2011] Anily, S., Gendreau, M., and Laporte, G. (2011). The preemptive swapping problem on a tree. *Networks*, 58(2) :83–94.
- [Anily and Hassin, 1992] Anily, S. and Hassin, R. (1992). The swapping problem. *Networks*, 22(4) :419–433.
- [Attanasio et al., 2004] Attanasio, A., Cordeau, J.-F., Ghiani, G., and Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3) :377–387.
- [Bent and Van Hentenryck, 2004a] Bent, R. and Van Hentenryck, P. (2004a). Regrets only! online stochastic optimization under time constraints. In *AAAI*, volume 4, pages 501–506.

-
- [Bent and Van Hentenryck, 2004b] Bent, R. and Van Hentenryck, P. (2004b). The value of consensus in online stochastic scheduling. In *ICAPS*, volume 4, pages 219–226.
- [Bent and Van Hentenryck, 2004c] Bent, R. W. and Van Hentenryck, P. (2004c). Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6) :977–987.
- [Berbeglia et al., 2007] Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems : a classification scheme and survey. *Top*, 15(1) :1–31.
- [Bertsimas, 1988] Bertsimas, D. (1988). *Probabilistic combinatorial optimization problems*. PhD thesis, Massachusetts Institute of Technology.
- [Bianchessi and Righini, 2007] Bianchessi, N. and Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, 34(2) :578–594.
- [Birattari et al., 2008] Birattari, M., Balaprakash, P., Stützle, T., and Dorigo, M. (2008). Estimation-based local search for stochastic combinatorial optimization using delta evaluations : a case study on the probabilistic traveling salesman problem. *INFORMS Journal on Computing*, 20(4) :644–658.
- [Borndörfer et al., 1999] Borndörfer, R., Grötschel, M., Klostermeier, F., and Küttner, C. (1999). Telebus berlin : Vehicle scheduling in a dial-a-ride system. In *Computer-Aided Transit Scheduling*, pages 391–422. Springer.
- [Brandao, 2006] Brandao, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, 173(2) :540–555.
- [Bredström and Rönnqvist, 2008] Bredström, D. and Rönnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European journal of operational research*, 191(1) :19–31.
- [Carrabs et al., 2007] Carrabs, F., Cordeau, J.-F., and Laporte, G. (2007). Variable neighborhood search for the pickup and delivery traveling salesman problem with lifo loading. *INFORMS Journal on Computing*, 19(4) :618–632.
- [Chalasani and Motwani, 1999] Chalasani, P. and Motwani, R. (1999). Approximating capacitated routing and delivery problems. *SIAM Journal on Computing*, 28(6) :2133–2149.

- [Chang et al., 2000] Chang, H. S., Givan, R., and Chong, E. K. (2000). On-line scheduling via sampling. In *AIPS*, pages 62–71.
- [Chiang and Russell, 1996] Chiang, W.-C. and Russell, R. A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63(1) :3–27.
- [Cordeau, 2006] Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3) :573–586.
- [Croes, 1958] Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, 6(6) :791–812.
- [Czech and Czarnas, 2002] Czech, Z. J. and Czarnas, P. (2002). Parallel simulated annealing for the vehicle routing problem with time windows. In *Proceedings 10th Euromicro workshop on parallel, distributed and network-based processing*, pages 376–383. IEEE.
- [Dell’Amico et al., 2006] Dell’Amico, M., Righini, G., and Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation science*, 40(2) :235–247.
- [Desrosiers et al., 1986] Desrosiers, J., Dumas, Y., and Soumis, F. (1986). A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences*, 6(3-4) :301–325.
- [Dohn et al., 2009] Dohn, A., Kolind, E., and Clausen, J. (2009). The manpower allocation problem with time windows and job-teaming constraints : A branch-and-price approach. *Computers & Operations Research*, 36(4) :1145–1157.
- [Dohn et al., 2011] Dohn, A., Rasmussen, M. S., and Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, 58(4) :273–289.
- [Eveborn et al., 2006] Eveborn, P., Flisberg, P., and Rönnqvist, M. (2006). Laps care-an operational system for staff planning of home care. *European journal of operational research*, 171(3) :962–976.
- [Flatberg et al., 2007] Flatberg, T., Hasle, G., Kloster, O., Nilssen, E. J., and Riise, A. (2007). Dynamic and stochastic vehicle routing in practice. In *Dynamic Fleet Management*, pages 41–63. Springer.

- [Frederickson and Guan, 1993] Frederickson, G. N. and Guan, D. J. (1993). Nonpreemptive ensemble motion planning on a tree. *Journal of Algorithms*, 15(1) :29–60.
- [Gambardella et al., 2003] Gambardella, L. M., Rizzoli, A. E., Oliverio, F., Casagrande, N., Donati, A., Montemanni, R., and Lucibello, E. (2003). Ant colony optimization for vehicle routing in advanced logistics systems. In *Proceedings of the International Workshop on Modelling and Applied Simulation (MAS 2003)*, pages 3–9.
- [Gendreau et al., 1992] Gendreau, M., Hertz, A., and Laporte, G. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6) :1086–1094.
- [Gendreau et al., 1996] Gendreau, M., Hertz, A., and Laporte, G. (1996). The traveling salesman problem with backhauls. *Computers & Operations Research*, 23(5) :501–508.
- [Gendreau et al., 1999] Gendreau, M., Laporte, G., and Vigo, D. (1999). Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research*, 26(7) :699–714.
- [Golden et al., 1980] Golden, B., Bodin, L., Doyle, T., and Stewart Jr, W. (1980). Approximate traveling salesman algorithms. *Operations research*, 28(3-part-ii) :694–711.
- [Gribkovskaia et al., 2001] Gribkovskaia, I., Halskau, O., and Myklebost, K. N. B. (2001). Models for pick-up and deliveries from depots with lasso solutions. *NOFOMA2001, Collaboration in logistics : connecting islands using information technology*, pages 279–293.
- [Gribkovskaia et al., 2007] Gribkovskaia, I., Halskau sr, Ø., Laporte, G., and Vlček, M. (2007). General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 180(2) :568–584.
- [Haghani and Jung, 2005] Haghani, A. and Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & operations research*, 32(11) :2959–2986.
- [Healy and Moll, 1995] Healy, P. and Moll, R. (1995). A new extension of local search applied to the dial-a-ride problem. *European Journal of Operational Research*, 83(1) :83–104.
- [Hernández-Pérez, 2004] Hernández-Pérez, H. (2004). Traveling salesman problems with pickups and deliveries. *Disertation, University of La Laguna, Spain*.

-
- [Hernández-Pérez and Salazar-González, 2004a] Hernández-Pérez, H. and Salazar-González, J.-J. (2004a). A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics*, 145(1) :126–139.
- [Hernández-Pérez and Salazar-González, 2004b] Hernández-Pérez, H. and Salazar-González, J.-J. (2004b). Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. *Transportation Science*, 38(2) :245–255.
- [Hoff et al., 2009] Hoff, A., Gribkovskaia, I., Laporte, G., and Løkketangen, A. (2009). Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 192(3) :755–766.
- [Hoff and Løkketangen, 2006] Hoff, A. and Løkketangen, A. (2006). Creating lasso-solutions for the traveling salesman problem with pickup and delivery by tabu search. *Central European Journal of Operations Research*, 14(2) :125–140.
- [Jaw et al., 1986] Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., and Wilson, N. H. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B : Methodological*, 20(3) :243–257.
- [Johnson and McGeoch, 1997] Johnson, D. S. and McGeoch, L. A. (1997). The traveling salesman problem : A case study in local optimization. *Local search in combinatorial optimization*, 1(1) :215–310.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598) :671–680.
- [Larsen et al., 2004] Larsen, A., Madsen, O. B., and Solomon, M. M. (2004). The a priori dynamic traveling salesman problem with time windows. *Transportation Science*, 38(4) :459–472.
- [Li et al., 2009a] Li, J.-Q., Mirchandani, P. B., and Borenstein, D. (2009a). A lagrangian heuristic for the real-time vehicle rescheduling problem. *Transportation Research Part E : Logistics and Transportation Review*, 45(3) :419–433.
- [Li et al., 2009b] Li, J.-Q., Mirchandani, P. B., and Borenstein, D. (2009b). Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research*, 194(3) :711–727.
- [Lin, 1965] Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10) :2245–2269.

-
- [Liu et al., 2019] Liu, R., Tao, Y., and Xie, X. (2019). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. *Computers & Operations Research*, 101 :250–262.
- [Madsen et al., 1995] Madsen, O. B., Ravn, H. F., and Rygaard, J. M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60(1) :193–208.
- [Mladenović and Hansen, 1997] Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11) :1097–1100.
- [Montané and Galvao, 2006] Montané, F. A. T. and Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3) :595–619.
- [Montemanni et al., 2005] Montemanni, R., Gambardella, L. M., Rizzoli, A. E., and Donati, A. V. (2005). Ant colony system for a dynamic vehicle routing problem. *Journal of combinatorial optimization*, 10(4) :327–343.
- [Mosheiov, 1994] Mosheiov, G. (1994). The travelling salesman problem with pick-up and delivery. *European Journal of Operational Research*, 79(2) :299–310.
- [Novoa and Storer, 2009] Novoa, C. and Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European journal of operational research*, 196(2) :509–515.
- [Or, 1977] Or, I. (1977). Traveling salesman type combinatorial problems and their relation to the logistics of regional blood banking.
- [Pillac et al., 2013] Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1) :1–11.
- [Psaraftis, 1980] Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2) :130–154.
- [Psaraftis, 1983] Psaraftis, H. N. (1983). Analysis of an $O(n^2)$ heuristic for the single vehicle many-to-many euclidean dial-a-ride problem. *Transportation Research Part B : Methodological*, 17(2) :133–145.

- [Rabeh et al., 2011] Rabeh, R., Saïd, K., and Eric, M. (2011). Collaborative model for planning and scheduling caregivers' activities in homecare. *IFAC Proceedings Volumes*, 44(1) :2877–2882.
- [Rekiek et al., 2006] Rekiek, B., Delchambre, A., and Saleh, H. A. (2006). Handicapped person transportation : An application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence*, 19(5) :511–520.
- [Salazar-Aguilar et al., 2012] Salazar-Aguilar, M. A., Langevin, A., and Laporte, G. (2012). Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, 39(7) :1432–1440.
- [Secomandi, 2000] Secomandi, N. (2000). Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, 27(11-12) :1201–1225.
- [Stein, 1978] Stein, D. M. (1978). An asymptotic, probabilistic analysis of a routing problem. *Mathematics of Operations Research*, 3(2) :89–101.
- [Süral and Bookbinder, 2003] Süral, H. and Bookbinder, J. H. (2003). The single-vehicle routing problem with unrestricted backhauls. *Networks : An International Journal*, 41(3) :127–136.
- [Van Breedam, 1995] Van Breedam, A. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research*, 86(3) :480–490.
- [Van der Bruggen et al., 1993] Van der Bruggen, L., Lenstra, J. K., and Schuur, P. (1993). Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science*, 27(3) :298–311.
- [Wade and Salhi, 2002] Wade, A. and Salhi, S. (2002). An investigation into a new class of vehicle routing problem with backhauls. *Omega*, 30(6) :479–487.
- [Yano et al., 1987] Yano, C. A., Chan, T. J., Richter, L. K., Cutler, T., Murty, K. G., and McGettigan, D. (1987). Vehicle routing at quality stores. *Interfaces*, 17(2) :52–63.