N° d'Ordre : D.U. 1990
EDSPIC : 468

Université Blaise Pascal - Clermont-Ferrand II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

THÈSE

présentée par

Mathieu LACROIX

pour obtenir le grade de

Docteur d'université
Spécialité : Informatique

Le problème de ramassage et livraison préemptif : complexité, modèles et polyèdres.

Soutenue publiquement le 07 décembre 2009 devant le jury :

A. R.	Mahjoub	Directeur de thèse
A.	Quilliot	Directeur de thèse
H.	Kerivin	Co-encadrant de thèse
J.-F.	Maurras	Président de jury
M.	Gendreau	Rapporteur
M.	Haouari	Rapporteur
F.	Semet	Rapporteur

Remerciements

Je voudrais en premier lieu remercier Messieurs A. Ridha Mahjoub et Alain Quillot de m'avoir permis d'effectuer une thèse sous leurs directions, et Monsieur Hervé Kerivin d'avoir accepté d'être co-encadrant sur cette thèse.

Je remercie tout particulièrement Monsieur A. Ridha Mahjoub, Professeur à l'Université Paris-Dauphine, de m'avoir apporté son soutien, scientifique et moral, tout au long de ce travail. Il a su me transmettre les connaissances, la rigueur, la motivation et la passion pour la recherche en général et l'optimisation combinatoire en particulier. Il m'a également permis de surmonter les moments difficiles grâce à son très grand intérêt pour mon travail et sa constante disponibilité.

Je remercie également vivement Monsieur Hervé Kerivin, professeur Assistant à l'Université de Clemson, pour toute l'aide qu'il m'a apportée durant cette thèse. Cette aide se situe bien sûr au niveau scientifique, grâce aux nombreuses discussions que nous avons eues sur mon travail et à ses conseils et idées. Mais il m'a également appris, grâce à sa haute exigence scientifique et à sa très grande estime de mes capacités, à ne jamais renoncer et à donner le meilleur de moi-même.

Je tiens aussi à remercier très sincèrement Monsieur Alain Quillot, Professeur à l'Université Blaise-Pacal de Clermont-Ferrand, pour l'enthousiasme et le dynamisme dont il a fait preuve tout au long de cet encadrement. Je tiens également beaucoup à le remercier pour ses idées et sa détermination qui m'ont permis d'élargir mon domaine de connaissances et le travail réalisé durant cette thèse.

J'ai été très honoré que Monsieur Michel Gendreau, chercheur au Centre de Recherche sur les Transports (Canada), ait accepté de rapporter ma thèse. Je le remercie pour l'attention toute particulière qu'il a portée à mon mémoire.

Je remercie également Monsieur Mohamed Haouari, Professeur à l'École Polytechnique de Tunisie, pour l'intérêt qu'il a bien voulu porter à ce travail et pour m'avoir fait l'honneur d'accepter la charge de rapporteur.

Ma gratitude va ensuite à Monsieur Frédéric Semet, Professeur à l'École Centrale de Lille, qui m'a fait l'honneur de rapporter ma thèse.

Mes remerciements vont également à Monsieur Jean-François Maurras, Professeur à l'Université de la Méditerranée de Marseille, pour avoir accepté d'examiner mes travaux et de présider le jury.

Je tiens également à remercier très chaleureusement les différents collègues et amis du LIMOS et du LAMSADE qui ont partagé avec moi ces différentes années. Les discussions animées et les moments de détente partagés, leur bonne humeur, leurs encouragements et leur amitié, mais aussi les échanges scientifiques et les travaux effectués ensemble, ont grandement contribué à mon épanouissement scientifique et personnel lors de ces années de thèse. Je tiens à remercier tout d'abord Madame Fatiha Bendali-Mailfert et Monsieur Jean Mailfert pour leur gentillesse et leurs encouragements. Je souhaite également remercier fortement Sylvie Borne, avec qui j'ai partagé beaucoup de bons moments et qui m'a beaucoup aidé, notamment au niveau administratif et par sa relecture de ce manuscrit. Merci également à Pierre Fouilhoux, Pierre Pesneau, Ibrahim Diarrassouba, Lise Slama, Jean-Christophe Gay et Hélène Toussaint qui ont partagé mon bureau et les bureaux voisins au LIMOS. Un grand merci aussi à Hélène pour sa collaboration concernant l'heuristique. Je tiens également à remercier très chaleureusement Denis Cornaz, pour tous les excellents moments passés ensemble et d'une manière plus générale, pour son amitié. Je veux aussi remercier tous mes collègues qui ont partagé avec moi le bureau C605 de Paris-Dauphine et le bureau voisin. Je tiens tout particulièrement à exprimer mon affection à Nabila Remli et Wassila Ouerdane pour les moments passés et nos soutiens respectifs, à Raouia Taktak pour sa bonne humeur, à Fabien Badeig, Guillaume Ravilly-Abadie et Nicolas Boria, notamment pour toutes ces pauses partagées dehors. Je tiens aussi à remercier particulièrement Nabila, Guillaume et Sébastien Martin qui ont accepté de travailler avec moi et de m'associer à leurs différents travaux de recherche.

Enfin, mes remerciements ne seraient pas complets si je ne remerciais pas ma famille et mes amis, de Clermont-Ferrand et de Paris, qui m'ont supporté et appuyé toutes ces années et se sont réellement intéressés à mon travail et au domaine hautement incompréhensible de l'optimisation combinatoire en général. Je leur sais gré de n'avoir pas laissé tomber et surtout de m'avoir offert des moments de détente, de soutien, d'écoute et de réconfort. Je ne les nommerai pas, ils se reconnaîtront sans aucun doute. Enfin, je remercie sincèrement du fond du cœur Félicie, qui m'a toujours soutenu tout au long de cette thèse, à chaque instant, et qui me soutient encore maintenant.

Résumé

Dans cette thèse, nous nous intéressons à différents problèmes liés à l'optimisation des tournées des véhicules. Nous considérons plusieurs variantes du problème de ramassage et livraison préemptif, c'est-à-dire, lorsque les demandes peuvent être temporairement déchargées en n'importe quel nœud du réseau.

Dans un premier temps, nous considérons le problème de ramassage et livraison préemptif asymétrique lorsque la flotte est composée de plusieurs véhicules, les transbordements sont possibles et le fractionnement de la demande est autorisé. Nous donnons deux formulations à l'aide de programmes linéaires mixtes basés sur un graphe spatio-temporel. Nous présentons des contraintes valides et fournissons, pour chacune des deux formulations, un algorithme de coupes et branchements pour résoudre le problème. Nous comparons finalement les résultats expérimentaux obtenus avec les deux algorithmes.

Nous considérons par la suite le problème de ramassage et livraison préemptif asymétrique lorsque la flotte est composée d'un unique véhicule et le fractionnement des demandes n'est pas autorisé. D'abord, nous étudions la structure des solutions du problème. Nous montrons qu'une solution doit être décrite par l'ensemble des arcs traversés par le véhicule, l'ordre dans lequel ces arcs sont traversés, et les arcs des chemins des demandes. Nous montrons aussi que ces informations sont nécessaires pour déterminer en temps polynomial si une solution est réalisable pour le problème. En effet, si une de ces informations manque, alors vérifier si une solution est réalisable est un problème NP-complet. Nous introduisons ensuite une formulation sous forme d'un programme linéaire en nombres entiers pour le problème. Nous nous intéressons également à la version unitaire de ce problème, c'est-à-dire, lorsque le véhicule ne transporte qu'une seule demande à la fois. Nous montrons que pour cette version, l'ordre dans lequel les arcs du véhicule sont traversés n'est pas nécessaire dans la description d'une solution. Nous donnons, pour cette version, une formulation sous forme d'un programme linéaire en nombres entiers. Nous étudions ensuite le polytope des solutions de ce problème. Nous caractérisons sa dimension et donnons des conditions nécessaires et suffisantes

pour que les contraintes de base du problème définissent des facettes. En utilisant ces résultats, nous développons un algorithme de coupes et branchements pour le problème. Nous étudions enfin le polytope associé dans la classe des circuits doublement orientés. Nous identifions des contraintes valides pour cette classe de graphes et donnons une description complète du polytope. Nous étudions aussi ce polytope dans les graphes décomposables par des opérations de 1-somme. Comme conséquence, nous donnons une description du polytope dans la classe des cactus.

En plus, nous considérons le problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire en supposant que toute instance vérifie les hypothèses suivantes : le graphe représentant le réseau est complet, chaque sommet est incident à au plus une demande et les coûts vérifient les inégalités triangulaires. Sous ces hypothèses, nous montrons que toute solution peut être maintenant décrite par l'ensemble des arcs traversés par le véhicule. Nous montrons aussi que, dans ce cas, le problème se ramène à la recherche d'un cactus de coût minimum vérifiant certaines propriétés. En se basant sur cette transformation, nous proposons une métaheuristique pour résoudre le problème.

Mots clés : Problème de ramassage et livraison préemptif, complexité, graphe, polytope, facette, séparation, algorithme de coupes et branchements, métaheuristique.

Abstract

In this thesis, we consider different variants of the preemptive pickup and delivery problem. The term preemptive means that demands may be temporarily unloaded during their transportation anywhere in the network.

We first consider the asymetric version of the problem when several vehicles are available and the demands may be split and carried with transshipments. We give two mixed-integer linear programming formulations for this problem that are based on a space-time graph. We present some valid inequalities and give for both formulations branch-and-cut algorithms to solve the problem. We also present some experimental results.

Afterwards, we focus on the case when only one vehicle is available and the demands cannot be split (but they can still be temporarily dropped anywhere). This problem, called the *Single-vehicle Preemptive Pickup and Delivery Problem*, is still NP-hard. We first study the structure of the solutions of the problem. We show that a solution has to be described by the set of arcs traversed by the vehicle, the order in which these arcs are traversed and the set of arcs of the demand paths. We also prove that these informations are necessary to check in polynomial time if a solution is feasible. This means that if one of these informations is missing, then checking if a solution is feasible is an NP-complete problem. Finally, we give an integer linear programming formulation for this problem. We also study this problem when only one demand is carried at the same time (unitary case). The problem remains NP-hard in this case. However, we show that the information concerning the sequence of arcs traversed by the vehicle is no more necessary to check in polynomial time if a solution is feasible. Indeed, in this case, the order on the arcs can be obtained from the sets of arcs traversed by the vehicle and the demands. We introduce a new formulation for this variant of the single-vehicle preemptive pickup and delivery problem based on these results.

Moreover, we study the facial structure of the associated polytope. We characterize its dimension and give necessary and sufficient conditions for the inequalities of the

formulation to define facets. Using these results, we develop a branch-and-cut algorithm for the problem. We also study the associated polytope when the digraph is reduced to a bi-oriented circuit. We show that in this case, further valid inequalities are necessary for describing the polytope. Furthermore, we show that if a digraph D is a 1-sum of two digraphs D_1 and D_2, then the polytope in D can be obtained from those in D_1 and D_2. As a consequence, we obtain a description of the polytope in the cacti.

We finally consider the problem when the digraph is complete, each node different from the depot is incident to at most one demand and the costs satisfy the triangle inequalities. We then show that in this case, the problem reduces to finding a minimum-cost cactus satisfying some conditions. Using this transformation, we devise a metaheuristic for the problem.

Keywords : preemptive pickup and delivery problem, complexity, graph, polytope, facet, separation problem, branch-and-cut algorithm, metaheuristic.

Table des matières

Introduction

Le transport de biens prend une place de plus en plus importante dans l'économie. Ceci est dû à l'augmentation du volume des échanges et des distances sur lesquelles ces marchandises sont transportées, résultant de la délocalisation et de l'ouverture des marchés. Afin de rester compétitif et dans l'idée d'une économie soutenable, il est important de réduire les coûts et la pollution engendrés par ce transport. Cette réduction des coûts et de la pollution peut notamment se faire par l'amélioration des trajets des véhicules transportant ces marchandises. Ceci revient alors à optimiser les trajets des véhicules. Ces problèmes d'optimisation de tournées de véhicules sont connus difficiles. Parmi ces derniers existe le problème de ramassage et livraison [16, 24, 76, 87]. Celui-ci consiste à minimiser le trajet des véhicules transportant des demandes de leur origine à leur destination. Dans sa version préemptive, les demandes peuvent être temporairement déchargées en n'importe quel nœud du réseau. Dans cette thèse, nous nous intéressons à cette version du problème.

Le problème de ramassage et livraison est un problème NP-difficile qui a été largement étudié dans la littérature. Par contre, la version préemptive n'a eu que très peu d'attention. Le but de cette thèse est d'étudier ce problème d'un point de vue polyédral et algorithmique.

L'approche polyédrale s'est avérée très puissante pour aborder des problèmes combinatoires difficiles. Cette technique, initiée par Edmonds [33] dans le cadre du problème de couplage, consiste à ramener le problème en question à la résolution d'un programme linéaire (ou d'une séquence de programmes linéaires), par la description complète (ou partielle) de son polytope des solutions par un système d'inégalités linéaires. Cette approche a été appliquée avec succès à plusieurs problèmes d'optimisation combinatoire comme le problème du voyageur de commerce, le problème de la coupe maximale et le problème de conception de réseau fiable. L'objectif de cette thèse est de développer des algorithmes se basant sur cette approche pour certaines variantes du problème de ramassage et livraison préemptif.

Le manuscrit est organisé comme suit. Dans le premier chapitre, nous introduisons quelques notions de base de graphe, polyèdres et de complexité et les notations qui seront utiles tout au long de la thèse. Le chapitre 2 présente le problème de ramassage et livraison. Cette présentation inclut les cas où les demandes peuvent être transportées avec rechargement ou fractionnement. Un état de l'art est donc donné pour chacune de ces variantes. Le chapitre 3 est consacré à l'étude du problème de ramassage et livraison préemptif asymétrique lorsque la flotte est composée de plusieurs véhicules, les transbordements sont possibles et le fractionnement de la demande est autorisé. Le chapitre 4 est centré sur le problème de ramassage et livraison préemptif asymétrique lorsque la flotte est composée d'un unique véhicule et le fractionnement des demandes n'est pas autorisé. Deux variantes de ce problème sont ainsi considérées, la version unitaire, quand une seule demande peut être transportée à la fois, et la version non unitaire. Des formulations en nombres entiers pour ces deux variantes du problème sont présentées. Nous étudions, dans le chapitre 5, le polytope des solutions du problème dans le cas unitaire. Nous proposons un algorithme de coupes et branchements pour résoudre le problème dans ce cas. Finalement, le chapitre 6 est consacré à l'étude du problème lorsque l'instance vérifie les hypothèses suivantes : le graphe est complet, chaque sommet est incident à au plus une demande et les coûts vérifient les inégalités triangulaires. Une nouvelle formulation est donnée et une métaheuristique est développée pour résoudre cette variante du problème.

Chapitre 1

Notions préliminaires

Dans ce chapitre, nous donnons quelques notions de base sur les polyèdres combinatoires et la théorie de la complexité. Nous effectuons ensuite un bref rappel sur les méthodes de coupes et branchements et les méthodes de recherche locale. Nous terminons ce chapitre par les notations et définitions qui seront utilisées tout au long de ce manuscrit.

1.1 Problèmes d'optimisation combinatoire

L'*Optimisation Combinatoire* est une des branches de l'informatique et des mathématiques appliquées. Elle concerne les problèmes pouvant se formuler de la façon suivante : soit $E = \{e_1, \ldots, e_n\}$ un ensemble fini appelé *ensemble de base*, où chaque élément e_i possède un *poids* $c(e_i)$. Soit $\mathcal{S}$ une famille de sous-ensembles de E. Si $S \in \mathcal{S}$, alors $c(S) = \sum_{e_i \in S} c(e_i)$ est le poids de S. Le problème consiste à déterminer un élément de $\mathcal{S}$, ayant le plus petit (ou le plus grand) poids. Un tel problème est appelé un *problème d'optimisation combinatoire*. L'ensemble $\mathcal{S}$ est appelé l'*ensemble des solutions* du problème.

Le mot *combinatoire*, qui désigne la discipline des mathématiques concernée par les structures discrètes ou finies, évoque l'existence d'une structure sous-jacente discrète (généralement un graphe). L'ensemble de solutions $\mathcal{S}$ est défini dans cette structure et peut avoir un nombre exponentiel d'éléments. Le mot *optimisation* signifie que l'on recherche le meilleur élément de l'ensemble de solutions $\mathcal{S}$.

De nombreux problèmes peuvent se formuler comme des problèmes d'optimisation

combinatoire, comme par exemple le problème du sac à dos, du voyageur de commerce, de conception de réseaux, de transport, de localisation,... L'optimisation combinatoire se trouve au carrefour de la théorie des graphes, de la programmation linéaire et de la programmation linéaire en nombres entiers. Elle est très liée à la théorie de la complexité d'algorithmes.

1.2 Rappels sur la théorie de la complexité

La théorie de la complexité est basée sur les travaux d'Edmonds [32] et de Cook [23]. Elle permet de classer un problème donné parmi les problèmes faciles ou difficiles. Dans cette section, nous rappelons quelques éléments de base. La théorie de la NP-complétude est présentée en détail dans Garey et Johnson [42].

Un *problème* est une question générale possédant des paramètres dont la valeur n'est pas connue. Un problème est décrit en donnant : une description générale de tous les paramètres, et une énumération des propriétés que la solution doit satisfaire. Une *instance* d'un problème est obtenue en spécifiant la valeur de chaque paramètre du problème. Un *algorithme* de résolution d'un problème donné est une procédure, décomposable en opérations élémentaires, qui pour chaque instance du problème, produit une solution. La *taille* d'un problème reflète le nombre de données nécessaires pour décrire une instance.

Un algorithme est dit en $O(f(n))$ s'il existe un scalaire c et un entier n_0 tels que nombre d'opérations élémentaires nécessaires pour résoudre une instance de taille n est au plus $c.f(n)$ pour tout $n \geq n_0$. Si f est une fonction polynomiale, alors l'algorithme est dit *polynomial*. Un problème appartient à la *classe* P s'il existe, pour toute instance du problème, un algorithme polynomial en la taille de l'instance permettant de résoudre le problème. Les problèmes de la classe P sont dits *faciles*.

Un *problème de décision* est un problème ayant deux réponses possibles : *oui* ou *non*. Soient $\mathcal{P}$ un problème de décision et $\mathcal{I}$ les instances de ce problème pour lesquelles la réponse est oui. $\mathcal{P}$ appartient à la *classe NP* (Nondeterministic Polynomial) s'il existe un algorithme polynomial qui permet de vérifier que la réponse est oui pour toute instance de $\mathcal{I}$. Il est clair que la classe P est contenue dans la classe NP. La différence entre P et NP n'a pas été prouvée, mais la conjecture est considérée comme hautement probable.

Nous distinguons également dans la classe NP, la classe des problèmes NP-complets. La NP-complétude s'appuie sur la notion de réduction polynomiale. Un problème de

décision P_1 se réduit polynomialement en un problème de décision P_2 s'il existe une fonction polynomiale f telle que, pour toute instance I de P_1, la réponse est oui si et seulement si la réponse de $f(I)$ pour P_2 est oui. Nous noterons alors $P_1 \alpha P_2$. Un problème $\mathcal{P}$ est *NP-complet*, s'il appartient à la classe NP et s'il existe un problème Q connu comme étant NP-complet tels que $Q \alpha P$. Cook a été le premier à montrer la NP-complétude d'un problème, celui de la satisfiabilité [23].

À tout problème d'optimisation combinatoire peut être associé un problème de décision. Tout problème d'optimisation combinatoire dont le problème de décision associé est NP-complet est dit *NP-difficile*.

1.3 Éléments de la théorie des polyèdres

Dans cette section, nous introduisons quelques définitions et propriétés de la théorie des polyèdres. Pour plus de détails, on peut se référer à Pulleyblanck [80] et Schrijver [89].

Soit $x \in \mathbb{R}^n$. On dit que x est une *combinaison linéaire* des points $x_1, ..., x_k \in \mathbb{R}^n$ s'il existe k scalaires $\lambda_1, \lambda_2, ..., \lambda_k \in \mathbb{R}$ tels que $x = \sum_{i=1}^{k} \lambda_i x_i$. Si, de plus, $\sum_{i=1}^{k} \lambda_i = 1$ alors on dit que x est une *combinaison affine* de ces points. De même, si $\lambda_i \geq 0 \, \forall \, i \in \{0, ..., k\}$ avec $\sum_{i=1}^{k} \lambda_i = 1$, alors on dit que x est une *combinaison convexe* de ces points.

Des points $x_1, ..., x_k \in \mathbb{R}$ sont dits *linéairement indépendants* (resp. *affinement indépendants*) si le système

$$\sum_{i=1}^{k} \lambda_i x_i = 0$$

$$(\text{resp.} \sum_{i=1}^{k} \lambda_i x_i = 0 \text{ et } \sum_{i=1}^{k} \lambda_i = 0)$$

admet une solution unique, $\lambda_i = 0, \quad \forall \, i = 1, ..., k$.

Soit S un ensemble non vide de points de $\mathbb{R}^n$. L'*enveloppe convexe* des points de S, notée *conv(S)*, est l'ensemble des points de $\mathbb{R}^n$ qui peuvent s'écrire comme combinaison convexe de points de S.

Un *polyèdre* P est un ensemble de points de $\mathbb{R}^n$ engendré par l'intersection d'un nombre fini de demi-espaces de $\mathbb{R}^n$. D'une manière équivalente, P est l'ensemble des solutions d'un système d'inégalités linéaires, c'est-à-dire $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, où

A est une matrice à m lignes et n colonnes, et b un vecteur de m composantes. Nous pouvons alors dire que le système $Ax \leq b$ caractérise le polyèdre P.

Un *polytope* est un polyèdre borné. Ainsi, un polyèdre $P \subseteq \mathbb{R}^n$ est un polytope si et seulement s'il existe $l, u \in \mathbb{R}^n$ tels que $l \leq x \leq u$, pour tout $x \in P$. Par exemple, l'enveloppe convexe d'un nombre fini de points est un polytope.

Un polyèdre P de $\mathbb{R}^n$ est de *dimension* d, notée *dim(P)*, si le nombre maximum de points de P affinement indépendants est égal à $d + 1$. Un polyèdre est de *pleine dimension* si *dim(P)*$= n$.

Un point $x \in \mathbb{R}^n$ est un *point extrême* d'un polyèdre P s'il ne peut s'écrire comme combinaison convexe d'autres points de P.

Une inégalité (ou *contrainte*) $a^T x \leq \alpha$, où a est un vecteur à n composantes et α un scalaire, est dite *valide* pour un polyèdre P si elle est vérifiée par tous les points de P, soit $P \subseteq \{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}$. Étant donné un point $x^* \in P$, nous dirons que l'inégalité $a^T x \leq \alpha$ est *serrée* (resp. *lâche*) pour x^* si $a^T x^* = \alpha$ (resp. $a^T x^* < \alpha$). Soit $a^T x \leq \alpha$ une contrainte valide pour le polyèdre P. Alors le sous-ensemble $F \subseteq P$ défini par $F = \{x \in P \mid a^T x = \alpha\}$ est une *face* de P. On dit alors que la face F est définie (ou induite) par la contrainte $a^T x \leq \alpha$. Si, de plus, $F \neq \emptyset$ et $F \neq P$ alors F est une *face propre*. Si F est propre et que $dim(F) = dim(P) - 1$, alors F est une *facette*.

Une inégalité est dite *essentielle* pour un polyèdre P si elle induit une facette pour P. Une inégalité est dite *redondante* dans un système $Ax \leq b$ définissant un polyèdre P, si le sous-système obtenu à partir de $Ax \leq b$ en supprimant cette inégalité définit le même polyèdre P.

1.4 Approche polyédrale, méthode de coupes et branchements

Soient $\mathcal{P}$ un problème d'optimisation combinatoire et $\mathcal{S}$ l'ensemble de ses solutions. Le problème $\mathcal{P}$ s'écrit alors

$$\min \{cx \mid x \in \mathcal{S}\}$$

où c est un vecteur coût associé aux variables du problème. Considérons l'enveloppe convexe $conv(\mathcal{S})$ des solutions de $\mathcal{P}$. Le problème $\mathcal{P}$ est équivalent au programme linéaire

$$\min \{cx \mid x \in conv(\mathcal{S})\}$$

L'*approche polyédrale*, introduite par Edmonds [33] dans le cadre du problème de couplage, consiste à décrire le polyèdre $conv(\mathcal{S})$ par un système d'inégalités linéaires, ramenant ainsi le problème $\mathcal{P}$ à la résolution d'un programme linéaire. Ceci permet donc de résoudre le problème $\mathcal{P}$ en utilisant les algorithmes de la programmation linéaire [26, 54, 55]. Rappelons que la résolution d'un programme linéaire peut être effectuée en temps polynomial [54, 55].

Une description complète du polyèdre peut cependant comporter un nombre exponentiel d'inégalités. Le problème d'optimisation sur le polytope $conv(\mathcal{S})$ ne peut donc être résolu comme un programme linéaire ayant explicitement toutes ses contraintes. Cependant, un nombre réduit de ces inégalités peut être suffisant pour résoudre le problème à l'aide d'une *méthode de coupes*. Cette méthode s'appuie sur le problème suivant, appelé *problème de séparation*.

Problème Soit $\mathcal{C}$ une classe d'inégalités valides pour $conv(\mathcal{S})$. Le problème de séparation associé à $\mathcal{C}$ consiste, étant donné un point $x \in \mathbb{R}^n$, à décider si x satisfait toutes les inégalités de $\mathcal{C}$ et sinon, à trouver une inégalité de $\mathcal{C}$ violée par x.

Le problème de séparation est un des points essentiels de l'approche polyédrale. En effet, Grötschel, Losvàsz et Schrijver [44] ont montré qu'un problème d'optimisation combinatoire sur un ensemble de contraintes $\mathcal{C}$ peut être résolu en temps polynomial si et seulement si le problème de séparation associé à $\mathcal{C}$ peut être résolu en temps polynomial. Ainsi, une méthode de coupes permet de résoudre un problème d'optimisation combinatoire en temps polynomial si l'on sait résoudre en temps polynomial le problème de séparation pour un système d'inégalités caractérisant le polyèdre associé. En effet, la méthode de coupes permet de résoudre un problème $\mathcal{P}$ comme une séquence de programmes linéaires, chacun contenant un nombre raisonnable de contraintes. Pour cela, la méthode commence par résoudre un programme linéaire contenant un sous-ensemble de contraintes de $conv(\mathcal{S})$. Notons par x^* la solution optimale obtenue. En appliquant les problèmes de séparation associés aux différentes classes d'inégalités définissant $conv(\mathcal{S})$, il est possible de savoir si x^* vérifie toutes les contraintes de $conv(\mathcal{S})$, *i.e.*, si $x^* \in conv(\mathcal{S})$. Si tel est le cas, alors x^* est la solution du problème. Autrement, les contraintes violées par x^* générées lors de l'application des problèmes de séparation sont ajoutées au programme linéaire. Ce processus de résolution est répété jusqu'à ce que la solution optimale appartienne à $conv(\mathcal{S})$.

Cependant, il est généralement difficile d'obtenir la caractérisation complète pour les problèmes NP-difficiles. De plus, le problème de séparation associé à certaines classes d'inégalités peut lui-même être NP-difficile. On ne peut dans ce cas obtenir que des techniques de séparation approchées. Ainsi, une méthode de coupes seule peut ne four-

nir que des solutions non réalisables (fractionnaires). Dans ce cas, on exécute une étape de branchement qui consiste à diviser le problème en plusieurs sous-problèmes tel que l'ensemble des solutions réalisables des sous-problèmes forme un recouvrement (idéalement une partition) des solutions du problème. Ce branchement peut, par exemple, se faire en choisissant une variable fractionnaire x_i et à considérer plusieurs (autant que le nombre de valeurs possibles pour x_i) sous-problèmes du problème courant en fixant dans chacun x_i à une de ses valeurs permises. On applique alors la méthode de coupes pour chacun des sous-problèmes. Ce processus continue jusqu'à l'obtention d'une solution optimale. Cette combinaison de la méthode de branchement et d'une méthode de coupes au niveau de chaque nœud de l'arbre est appelée *méthodes de coupes et branchements*.

Cette approche est largement utilisée pour les problèmes d'optimisation combinatoire.

1.5 Méthodes de recherche locale

Malgré l'efficacité des méthodes de résolution exactes, beaucoup de problèmes d'optimisation combinatoire ne peuvent être résolus de manière exacte que pour des instances de taille relativement faible. Il est donc intéressant dans ce cas d'utiliser des méthodes de résolution approchées. Bien que ces dernières ne donnent généralement pas la solution optimale du problème, elles peuvent néanmoins donner des solutions proches de l'optimum et permettent de résoudre des instances de grande taille. Parmi ces différentes méthodes existent les méthodes dites *méthodes de recherche locale* [57, 58]. Ces dernières consistent à visiter itérativement de proche en proche des solutions du problème.

Les méthodes de recherche locales sont définies par un *voisinage* et une *méthode de descente*. Le voisinage d'une solution définit l'ensemble des *solutions voisines*, considérées comme proches de la solution, qui peuvent être atteintes à chaque itération. Ce voisinage est défini par des *opérateurs de voisinage* qui modifient localement la solution courante. De nombreux opérateurs de voisinages existent et dépendent généralement du problème d'optimisation considéré. Parmi les opérateurs les plus connus figure le k-inter-échange développé pour le Problème du Voyageur de Commerce qui consiste à supprimer dans la solution courante (*i.e.*, un cycle hamiltonien) k arêtes pour les remplacer par k autres afin de former un nouveau cycle hamiltonien. La méthode de descente indique tout d'abord quelle solution voisine choisir : n'importe quelle solution voisine ou celle de meilleur coût par exemple. Elle permet aussi de décider si la solution courante doit être remplacée par la solution voisine ou non. La méthode de descente

simple consiste à remplacer la solution courante si cette dernière a un coût meilleur que la solution courante. Cependant, afin de sortir des optima locaux, il est possible d'accepter, dans certaines méthodes telles que la méthode tabou ou le recuit simulé, des solutions dégradées (*i.e.*, de moins bons coûts que la solution courante).

Les méthodes de recherche locale commencent avec une première solution, appelée *solution initiale*, qui sert au départ de solution courante. La méthode s'exécute ensuite de manière itérative. À chaque itération, l'algorithme choisit une solution voisine (suivant le voisinage de la méthode et la règle de choix d'une solution à l'intérieur de ce voisinage). Si la méthode de descente considère que la solution voisine est acceptée, cette dernière devient alors la solution courante. Dans tous les cas, l'heuristique passe à l'itération suivante. La fin de la méthode de recherche locale est donnée par des paramètres. Elle intervient généralement lorsque la solution courante n'a pas pu être améliorée pendant un certain nombre d'itérations ou après un certain temps d'exécution.

1.6 Notations et définitions

Nous donnons dans cette section certaines définitions et notions préliminaires de la théorie des graphes.

1.6.1 Graphes non orientés

Un *graphe* non orienté est noté $G = (V, E)$ où V est l'*ensemble des sommets* et E l'*ensemble des arêtes*.

Si e est une arête reliant deux sommets u et v, alors u et v seront appelés les *extrémités* de e, et nous écrirons $e = uv$ ou $e = \{u, v\}$. Si u est une extrémité de e, alors u (resp. e) est dit *incident* à e (resp. u). De même, deux sommets u et v formant une arête sont dits *adjacents*.

Si $F \subseteq E$ est un sous-ensemble d'arêtes, alors $V(F)$ représente l'ensemble des extrémités des arêtes de F. Si $W \subseteq V$ est un sous-ensemble de sommets, alors $E(W)$ denote l'ensemble des arêtes ayant leurs deux extrémités dans W.

Un *sous-graphe* $H = (U, F)$ de G est un graphe tel que $U \subseteq V$ et $F \subset E$. Un sous-graphe $H = (U, F)$ de G est dit *couvrant* si $U = V$. Soit $W \subseteq V$, $H = (W, E(W))$ est dit sous-graphe de G *induit* par W et sera noté par $G(W)$.

Si $F \subset E$ (resp. $W \subset V$), on notera par $G\backslash F$ (resp. $G\backslash W$) le graphe obtenu à partir de G en supprimant les arêtes de F (resp. les sommets de W et les arêtes incidentes à W). Si F (resp. W) est réduit à une seule arête e (resp. un seul sommet v), nous écrirons $G\backslash e$ (resp. $G\backslash v$).

Soit $W \subseteq V$, $\emptyset \neq W \neq V$, un sous-ensemble de sommets de V. L'ensemble des arêtes ayant une extrémité dans W et l'autre dans $V\backslash W$ est appelé *coupe* et noté $\delta(W)$. En posant $\overline{W} = V\backslash W$, nous avons $\delta(W) = \delta(\overline{W})$. Si W est réduit à un seul sommet v, nous écrirons $\delta(v)$. La cardinalité de la coupe $\delta(W)$ d'un sous-ensemble W est appelée *degré* de W et notée $\deg(W)$.

Étant donnés W et W' deux sous-ensembles disjoints de V, alors $[W, W']$ représente l'ensemble des arêtes de G qui ont une extrémité dans W et l'autre dans W'.

Si $\{V_1, \ldots, V_p\}$, $p \geq 2$, est une partition de V, alors $\delta(V_1, \ldots, V_p)$ est l'ensemble des arêtes $e = uv$ telles que $u \in V_i$, $v \in V_j$ et $i \neq j$.

Soient u et v deux sommets de V. Une *chaîne* P entre u et v est une séquence alternée de sommets et d'arêtes $(v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k)$ où $v_0 = u$, $v_k = v$, $e_i = v_{i-1}v_i$ pour $i = 1, \ldots, k$. P est dite *élémentaire* si elle passe au plus une fois par le même sommet (à l'exception de v_0 et v_k si ces derniers représentent le même sommet de G). Une chaîne élémentaire est donc totalement identifiée par son ensemble d'arêtes.

Deux chaînes entre u et v sont dites *arête-disjointes* (resp. *sommet-disjointes*) s'il n'existe pas d'arête (resp. de sommet différent de u et v) apparaissant dans les deux chaînes.

1.6.2 Graphes orientés

Un *graphe* orienté est noté $D = (V, A)$ où V est l'*ensemble de sommets* et A l'*ensemble des arcs*.

Si $a \in A$ est un arc reliant un sommet u à un sommet v, alors u sera appelé *extrémité initiale* et v *extrémité terminale* et nous écrirons $a = (u, v)$. On dit que a est un *arc sortant* de u et un *arc entrant* de v. Les sommets u et v sont appelés *extrémités* de a. Si v est une extrémité (initiale ou terminale) de a, alors v (resp. a) est dit *incident* à a (resp. v).

Si $B \subseteq A$ est un sous-ensemble d'arcs, alors $V(B)$ représente l'ensemble des extrémités des arcs de B. Si $W \subseteq V$ est un sous-ensemble de sommets, $A(W)$ représente l'ensemble des arcs ayant leurs deux extrémités dans W.

Un *sous-graphe* $H = (U, F)$ de D est un graphe tel que $U \subseteq V$ et $F \subset A$. Un sous-graphe $H = (U, F)$ de D est dit *couvrant* si $U = V$.

Si $F \subset A$ (resp. $W \subset V$), on notera par $D\backslash F$ (resp. $D\backslash W$) le graphe obtenu à partir de D en supprimant les arcs de F (resp. les sommets de W et les arêtes incidentes à W). Si F (resp. W) est réduit à un seul arc a (resp. un seul sommet v), nous écrirons $D\backslash a$ (resp. $D\backslash v$).

Soit $W \subseteq V$, $\emptyset \neq W \neq V$, un sous-ensemble de sommets de V. L'ensemble des arcs ayant leur extrémité initiale dans W et leur extrémité terminale dans $V\backslash W$ est appelé *coupe sortante* et noté $\delta^{\text{out}}(W)$. La cardinalité de la coupe sortante $\delta^{\text{out}}(W)$ d'un sous-ensemble W est appelée *degré sortant* de W et notée $\deg^{\text{out}}(W)$. Si $u \in W$ et $v \in V \setminus W$, alors la coupe sortante est aussi appelée *uv-coupe sortante*. Si W est réduit à un seul sommet v, nous écrirons repectivement $\delta^{\text{out}}(v)$ et $\deg^{\text{out}}(v)$ au lieu de $\delta^{\text{out}}(\{v\})$ et $\deg^{\text{out}}(\{v\})$.
L'ensemble des arcs ayant l'extrémité terminale dans W et l'extrémité initiale dans $V\backslash W$ est appelé *coupe entrante* et noté $\delta^{\text{in}}(W)$. La cardinalité de la coupe entrante $\delta^{\text{in}}(W)$ d'un sous-ensemble W est appelée *degré entrant* de W et notée $\deg^{\text{in}}(W)$. Si $u \in W$ et $v \in V \setminus W$, alors la coupe entrante est aussi appelée *uv-coupe entrante*. Si W est réduit à un seul sommet v, nous écrirons repectivement $\delta^{\text{in}}(v)$ et $\deg^{\text{in}}(v)$ au lieu de $\delta^{\text{in}}(\{v\})$ et $\deg^{\text{in}}(\{v\})$.
La *coupe* d'un ensemble $W \subseteq V$, $\emptyset \neq W \neq V$, est notée $\delta(W)$ et correspond à l'union des arcs de la coupe entrante et de la coupe sortante, *i.e.*, $\delta(W) = \delta^{\text{out}}(W) \cup \delta^{\text{in}}(W)$. La cardinalité de la coupe est appelée *degré* de W et notée $\deg(W)$. Si $u \in W$ et $v \in V\backslash W$, alors la coupe est aussi appelée *uv-coupe*. Si W est réduit à un seul sommet v, nous écrirons repectivement $\delta(v)$ et $\deg(v)$ au lieu de $\delta(\{v\})$ et $\deg(\{v\})$.
Si l'ensemble W associée à la coupe sortante $\delta^{\text{out}}(W)$ contient le sommet u mais pas le sommet v, on parle alors d'une *uv*-coupe sortante.

Étant donnés des sous-ensembles disjoints $W_1, W_2, \ldots, W_k$ de V, alors $[W_1, W_2, \ldots, W_k]$ représente l'ensemble des arcs de D qui ont une extrémité dans W_i et l'autre dans W_j, $i \neq j$.

Un graphe orienté $D = (V, A)$ est *faiblement connexe* si aucune coupe de D n'est vide. Le graphe D est dit *k-arc connexe* si $\deg^{in}(W) \geq k$ pour tout $W \subseteq V$, $\emptyset \neq W \neq V$. Un graphe 1-arc connexe est également appelé un graphe *fortement connexe*. Si D n'est pas faiblement connexe, alors chaque sous-graphe connexe (respectivement fortement connexe), maximal au sens de l'inclusion, de D est appelé *composante connexe* (respectivement *fortement connexe*). Un sommet $v \in V$ est appelé *sommet d'articulation* de D si le nombre de composantes connexes du graphe $D \setminus v$ contient est strictement

supérieur au nombre de composantes connexes de D. Un graphe $D = (V, A)$ est dit *Eulérien* s'il est connexe et si chaque sommet v de V vérifie $\deg^{\text{in}}(v) = \deg^{\text{out}}(v)$.

Soient D un graphe orienté et u et v deux sommets de V. Un *chemin* P de u à v dans D est une séquence d'arcs $((u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k))$ avec $u_1 = u$, $v_k = v$, $v_i = u_{i+1}$ pour tout $i = 1, 2, \ldots, k-1$ et $(u_i, v_i) \in A$ pour tout $i = 1, 2, \ldots, k$. Les sommets u et v sont respectivement appelés *sommet initial* et *sommet terminal* de P. Le nombre d'arcs de P est appelé *cardinalité* de P et noté $|P|$. P est appelé *circuit* si $u = v$. P est dit *élémentaire* si chaque sommet a au plus un arc entrant et un arc sortant parmi les arcs de P. Si P est un chemin élémentaire, alors P est défini par l'ensemble d'arcs apparaissant dans P. Si P est un circuit élémentaire, alors P est défini par l'ensemble d'arcs apparaissant dans P et par le sommet initial de P. Soit $Q = ((s_1, t_1), (s_2, t_2), \ldots, (s_l, t_l))$ un chemin dans D. Q est dit *sous-chemin* de P s'il existe $j \in \{1, 2, \ldots, k+1-l\}$ tel que $(s_i, t_i) = (u_{i+j}, v_{i+j})$ pour tout $i = 1, 2, \ldots, l$.

Si un graphe $D = (V, A)$ ne contient pas de circuit, alors D est dit *acyclique*. Si D est un graphe connexe acyclique tel que chaque chaque sommet a un degré entrant inférieur ou égal à un, alors D est appelé une r-arborescence, où r est le sommet de degré entrant égal à 0, le degré d'un sommet de $V \setminus \{r\}$ étant, par définition, égal à un. Si un sous-graphe $H = (U, F)$ de $D = (V, A)$ est une r-arborescence, alors on dit que H est une *r-arborescence de D couvrant U*. Par abus de langage, si le graphe $H = (V(F), F)$ induit par le sous-ensemble d'arcs $F \subseteq A$ est une r-arborescence de D couvrant $V(F)$, alors on dit que F est une r-arborescence de D couvrant $V(F)$. Si une r-arborescence de D couvre l'ensemble des sommets V, on parle alors d'une *r-arborescence de D couvrante*.

Chapitre 2

Le problème de ramassage et livraison

Dans ce chapitre, nous présentons un état de l'art sur le problème de ramassage et livraison sur lequel porte notre travail. Nous discutons de plusieurs variantes du problème étudiées dans la littérature. Nous présentons d'abord la version classique du problème de ramassage et livraison. Nous considérons ensuite les cas où les demandes peuvent respectivement être transportées avec rechargements ou fractionnement. Finalement, nous discutons d'une dernière variante dans laquelles les demandes sont transportées à la fois avec rechargements et fractionnement. Pour chaque variante, différents algorithmes de résolution seront discutés.

2.1 Présentation générale

L'effondrement du bloc soviétique a donné naissance à un nouveau système économique mondial, caractérisé par une politique libérale et une mondialisation des échanges. L'une des conséquences est l'augmentation notable du volume des marchandises échangées ainsi que la distance sur laquelle ces dernières sont transportées. Cette augmentation entraîne non seulement une élévation des coûts de production (de l'ordre de 10 à 15 %), due également à une hausse du prix du carburant, mais aussi une augmentation de la pollution. À ce titre, l'amélioration du transport des marchandises constitue un enjeu majeur pour les entreprises et les pouvoirs publics. De ce fait, les problèmes de transport constituent depuis plusieurs décennies un vaste domaine de recherche pour la communauté scientifique. Malgré la complexité de ces problèmes, cette dernière tente constamment de proposer de nouvelles méthodes de résolution exactes et heuristiques et de prendre en compte de plus en plus d'hypothèses et de contraintes

afin de répondre au mieux à ces questions de logistique.

L'un des problèmes les plus étudiés à ce jour est le problème de ramassage et livraison (pickup and delivery problem) qui consiste à ramasser des demandes à certains endroits pour les acheminer à leur destination à l'aide de véhicules. Ce problème s'appuie sur un réseau constitué d'un ensemble de nœuds et de liens. Les nœuds correspondent à certains lieux tels que des villes ou des régions. Les liens correspondent aux différentes routes pouvant être utilisées par les véhicules pour relier un lieu à un autre. On dispose également d'une flotte de véhicules, chacun étant défini par une capacité et un nœud dépôt. À chaque lien est associé un coût correspondant au coût généré lors du passage d'un véhicule sur ce lien. Un ensemble de paires de nœuds correspondant aux origines et destinations des demandes à transporter est également donné. À chaque demande est associé un volume, celui-ci représentant la quantité de la demande devant être transportée de son origine à sa destination. La version classique du problème de ramassage et livraison consiste à déterminer le trajet des véhicules et des demandes, où chaque véhicule commence et termine son trajet à son dépôt, chaque demande est transportée en une seule fois de son origine à sa destination à l'aide d'un unique véhicule et le coût total de transport est minimum. Afin d'obtenir des trajets réalisables, il faut s'assurer qu'à tout instant, la somme des volumes des demandes transportées par chaque véhicule ne dépasse pas sa capacité.

2.2 Modélisation en termes de graphes

Dans cette section, nous présentons une formulation du problème de ramassage et livraison en termes de graphes. Pour cela, nous considérons un graphe non-orienté $G = (V, E)$ réprésentant le réseau de transport. L'ensemble des sommets V correspond à l'ensemble des nœuds du réseaux et l'ensemble E à l'ensemble des liens pouvant être utilisés par les véhicules. Nous supposons que le graphe G est simple. À chaque arête $e \in E$ est associé un coût c_e correspondant au coût engendré par le passage d'un véhicule sur le lien e (les coûts sont symétriques).

Soit $K = \{1, 2, \ldots, p\}$, $p \geq 1$, un ensemble de demandes. À chaque demande $k \in K$ est associé un couple $(o^k, d^k) \in V \times V$, $o^k \neq d^k$, où o^k et d^k correspondent respectivement à l'origine et à la destination de la demande k. Les deux sommets origine et destination sont également appelés *extrémités* de la demande. Étant donné un sommet v de V, on dit que v est *incident* à une demande $k \in K$ si v est une des extrémités de k. On associe également à chaque demande $k \in K$ une valeur q^k correspondant au volume de la demande, *i.e.*, la quantité de demande devant être transportée de l'origine o^k à la

destination d^k.

Soit F une flotte de véhicules. Chaque véhicule $f \in F$ possède une capacité Q_f et a pour dépôt le sommet $w_f \in V$. Si tous les véhicules ont la même capacité et le même sommet de dépôt, on parle alors d'une flotte *homogène*. Dans le cas contraire, la flotte est dite *hétérogène*. Puisque chaque demande doit être transportée en une seule fois par un seul véhicule, il est clair que le volume de chaque demande doit être inférieur ou égal à la capacité maximum du véhicule, *i.e.*, $q^k \leq \max\{Q_f : f \in F\}$ pour tout $k = 1, 2, \ldots, p$.

Lorsque l'étude du problème de ramassage et livraison ne se limite pas à une classe de graphes particulière, on suppose, sans perte de généralité, que le graphe G est complet et les coûts vérifient les inégalités triangulaires, *i.e.*,

$$c_{uv} + c_{vw} - c_{uw} \geq 0 \qquad \forall u \neq v \neq w \neq u \in V \tag{2.1}$$

On suppose également, dans ce cas, que chaque sommet est incident à au plus une demande ou correspond à au plus un dépôt. De plus, aucune demande n'est incidente à un dépôt. Sous ces hypothèses, l'ensemble des sommets V n'est rien d'autre que l'ensemble formé des origines et destinations des demandes et des dépôts des véhicules. De plus, les trajets des demandes et des véhicules peuvent respectivement être représentés par des chaînes et des cycles élémentaires. En effet, les extrémités des demandes ne doivent être traversées qu'une seule fois par un unique véhicule et le plus court chemin entre deux sommets u et v est l'arête (u, v). Il est aussi facile de voir que l'ensemble des cycles élémentaires des véhicules couvre tous les sommets du graphe et que ces cycles sont sommet-disjoints. Finalement, puisque chaque demande est transportée sans interruption par un seul véhicule et les trajets des véhicules correspondent à des cycles élémentaires, on peut facilement déduire les chaînes élémentaires des demandes à l'aide des cycles élémentaires des véhicules. Une solution réalisable du problème de ramassage et livraison consiste donc en un ensemble de cycles élémentaires sommet-disjoints $C_1, C_2, \ldots, C_{|F|}$, où C_f, $f \in F$, est un cycle élémentaire passant par l'ensemble des sommets V_f et correspondant au trajet du véhicule f, tel que

1 - le cycle C_f passe par w_f pour tout $f \in F$,

2 - si un véhicule $f \in F$ traverse une extrémité d'une demande $k \in K$, alors les deux extrémités sont traversées par le véhicule, *i.e.*, o^k et d^k appartiennent à V_f,

3 - le chemin obtenu en parcourant le cycle C_f à partir du dépôt w_f (dans un sens ou dans l'autre) traverse l'origine o^k avant la destination d^k pour toute demande telle que $\{o^k, d^k\} \subseteq V_f$,

4 - la quantité transportée par le véhicule $f \in F$ ne dépasse jamais Q_f.

Le coût d'une solution $C_1, C_2, \ldots, C_{|F|}$ correspond à la somme des coûts des arêtes traversées par les véhicules, *i.e.*,

$$c(C_1, C_2, \ldots, C_{|F|}) = \sum_{f \in F} \sum_{e \in C_f} c_e.$$

Le *Problème de Ramassage et Livraison (PRL)* consiste alors à déterminer une solution $C_1, C_2, \ldots, C_{|F|}$ telle que le coût $c(C_1, C_2, \ldots, C_{|F|})$ soit minimum. Si la flotte de véhicules est constituée d'un unique (respectivement de plusieurs) véhicule(s), nous parlons alors du *PRL mono-véhicule (1-PRL)* (respectivement *PRL multi-véhicules (m-PRL)).*

Comme pour beaucoup d'autres problèmes de transport, il est possible que les coûts ne soient pas symétriques, *i.e.*, le coût pour traverser une arête uv de u vers v est différent de celui pour traverser uv de v vers u. Nous parlons alors de coûts asymétriques. Dans ce cas, le réseau doit alors être représenté par un graphe orienté et toute solution correspond alors à un ensemble de circuits élémentaires sommet-disjoints vérifiant les propriétés précédentes (la troisième propriété est légèrement modifiée car le seul sens de parcours du circuit possible est celui donné par l'orientation des arcs). Le problème dans ce cas est appelé *Problème de Ramassage et Livraison Asymétrique (PRLA).* Comme dans le cas symétrique, une distinction est faite suivant la taille de la flotte. Le problème est appelé *PRLA mono-véhicule (1-PRLA)* si la flotte est constituée d'un unique véhicule. Dans le cas contraire, on parle du *PRLA multi-véhicules (m-PRLA).*

2.3 État de l'art

Dans cette section, nous présentons un état de l'art de différents algorithmes utilisés pour résoudre le PRL et certaines variantes de ce problème. Comme de très nombreuses variantes de ce problème ont été étudiées, nous nous concentrons uniquement sur le cas statique, *i.e.*, l'information relative aux demandes est connue a priori (voir [17, 87] pour un état de l'art sur le cas dynamique). De même, nous ne considérons que la version du problème avec capacité de transport limitée pour les véhicules. En effet, comme notre travail porte sur les notions de rechargements et de fractionnement dans le transport des demandes et que ces notions sont intimement liées à la limitation de la capacité de transport, il paraît hors de propos de lister les travaux associés au cas sans capacité. La littérature portant sur le problème avec capacité infinie sera donc omise. Le lecteur pourra se référer à [16, 76, 87] pour un état de l'art sur les problèmes sans contraintes de capacité. Nous décrivons dans un premier temps différents algorithmes

utilisés pour résoudre la version classique du problème de ramassage et livraison, *i.e.*, sans rechargements ni fractionnement. Nous considérons par la suite trois variantes du problème de ramassage et livraison. Celles-ci correspondent respectivement aux cas où les demandes sont transportées avec rechargements, avec fractionnement, et avec rechargements et fractionnement. Pour chaque variante, nous présentons certaines méthodes de résolution.

2.3.1 Résolution du problème de ramassage et livraison

Dans cette partie, nous discutons du problème de ramassage et livraison lorsque les demandes sont transportées sans rechargements et ne peuvent être fractionnées, comme défini dans la section 2.2. Nous discutons de certaines méthodes exactes et heuristiques développées pour ce problème. Nous présenterons d'abord celles utilisées pour le problème sans contraintes de fenêtres de temps puis considérons le cas où ces contraintes sont prises en compte. Nous présentons également plusieurs cas particuliers et extensions de ce problème et donnons quelques méthodes de résolution pour ces problèmes. Cette présentation n'est pas exhaustive, pour une description plus complète, se référer à [16, 24, 76, 87].

2.3.1.1 PRL sans contraintes de fenêtres de temps

L'un des premiers travaux sur le problème de ramassage et livraison (avec contraintes de capacité) est celui de Psaraftis [78]. Dans ce travail, chaque demande correspond au déplacement d'une personne. La flotte est composée d'un unique véhicule pouvant transporter plusieurs personnes à la fois. Le coût correspond à une combinaison pondérée entre le temps de trajet total du véhicule et le temps d'attente et de transport de chaque client. Finalement, comme l'algorithme a vocation à être utilisé dans un contexte dynamique, les clients sont numérotés - cette numérotation correspondant à l'ordre d'apparition du client dans le cas dynamique. Afin que les clients n'attendent pas trop longtemps, des contraintes imposent que l'ordre dans lequel les personnes sont transportées ne diffère pas trop de l'ordre d'énumération. L'auteur a introduit un algorithme basé sur la programmation dynamique pour résoudre ce problème. Sa complexité est en $O(p^2 3^p)$ où p est le nombre de demandes.

Un autre algorithme exact pour le problème de ramassage et livraison mono-véhicule a également été proposé par Hernández-Pérez et Salazar-González [49]. Dans ce travail, les auteurs ont étudié le problème de ramassage et livraison asymétrique mono-véhicule

où les sommets peuvent être incidents à plusieurs demandes et toute solution doit traverser exactement une fois chaque sommet du graphe. Ils ont formulé ce problème à l'aide d'un programme linéaire mixte. Ce dernier est une formulation du *Problème du Voyageur de Commerce Asymétrique (PVCA)* dans laquelle des flots sont associés à chaque demande. Dans un premier modèle, les flots sont représentés selon une approche arc-sommets. Un deuxième modèle est donné en remplaçant les contraintes de flot par des contraintes métriques par application du lemme de Farkas. Finalement, un troisième modèle est donné en appliquant le lemme de Farkas pour l'existence d'un flot réalisable lorsque ce dernier est représenté par un modèle arc-chemins. Les auteurs ont introduit des contraintes valides pour les trois modèles qui sont utilisées pour résoudre le problème à l'aide d'un algorithme de coupes et branchements. Les résultats expérimentaux semblent indiquer que la meilleure approche est l'algorithme de coupes et branchements basé sur le deuxième modèle.

Finalement, une version proche du problème de ramassage et livraison mono-véhicule a été étudiée par Fiala Timlin et Pulleyblank [36, 37]. Dans cette version, les sommets sont partitionnés en plusieurs sous-ensembles. L'origine et la destination de chaque demande appartiennent au même sous-ensemble et chaque sommet peut être incident à plusieurs demandes. De plus, les sous-ensembles sont ordonnés par ordre de visite et les coûts correspondent aux distances euclidiennes. Le problème consiste à déterminer une solution du problème de ramassage et livraison dans laquelle les sous-ensembles sont visités dans l'ordre, *i.e.*, tous les sommets d'un sous-ensemble doivent être visités avant n'importe quel sommet d'un autre sous-ensemble ayant une priorité de visite plus faible. Les auteurs ont développé une heuristique pour résoudre ce problème. Pour chaque sous-ensemble de sommets, plusieurs problèmes de ramassage et livraison sont résolus à l'aide d'une recherche locale dont les solutions voisines sont déterminées à l'aide d'un 3-opt modifié ; la différence entre ces différents problèmes résidant dans le choix des sommets de départ et d'arrivée du véhicule. La combinaison des solutions obtenues pour tous les sous-ensembles forment des solutions du problème et la meilleure est déterminée par un algorithme du plus court chemin.

D'autres travaux dans la littérature portent sur le problème de ramassage et livraison multi-véhicules. Kalantari et al. [53] ont proposé un algorithme de séparation et évaluation pour résoudre le m-PRLA. Cet algorithme est basé sur celui défini par Little et al. [59] pour le problème du voyageur de commerce asymétrique. Dans cet algorithme, Little et al. déterminent à partir de la matrice initiale des coûts une seconde matrice, appelée matrice réduite, telle que le coût de chaque circuit hamiltonien donné par la matrice initiale est supérieur ou égal à celui donné par la matrice réduite plus une constante fixée (l'égalité est atteinte pour tout circuit hamiltonien optimal). L'évaluation à chaque nœud de l'arbre de branchements est déterminée par la somme des

constantes calculées pour chaque matrice depuis la racine jusqu'au nœud considéré. Des pénalités sont également ajoutées pour chaque arc interdit dans la solution. Le branchement consiste, étant donné un arc, à considérer un premier sous-problème dans lequel cet arc appartient au circuit hamiltonien et un deuxième dans lequel le circuit ne peut contenir cet arc (arc interdit). Durant l'algorithme, la matrice réduite est modifiée en ajoutant des coûts infinis associés aux arcs induisant des solutions non réalisables pour le problème (création de sous-circuits). Pour le 1-PRLA, Kalantari et al. ont étendu cet algorithme en ajoutant des coûts infinis pour les arcs induisant des solutions violant des contraintes de précédence ou de capacité. Le m-PRLA est transformé en un problème mono-véhicule en créant une copie du dépôt pour tout véhicule additionnel. Le circuit hamiltonien se découpe alors en autant de chemins que de véhicules, les extrémités de chaque chemin correspondant au dépôt ou aux copies. De nouveaux arcs sont exclus afin de s'assurer que chaque demande est transportée par un seul véhicule. Kalantari et al. ont cependant testé leur algorithme uniquement dans le cas d'un seul véhicule de capacité infinie.

Lu et Dessouky [62] ont étudié le problème de ramassage et livraison multi-véhicules lorsque les véhicules possèdent une capacité différente et le sommet de départ d'un véhicule correspond au sommet d'arrivée du véhicule précédent (les véhicules ne retournent donc pas à l'endroit où ils ont commencé leur trajet). Le coût total d'une solution correspond au coût fixe d'utilisation d'un véhicule et à la distance parcourue par tous les véhicules. Les auteurs ont transformé le problème multi-véhicules en un problème mono-véhicule dans lequel la capacité du véhicule varie suivant le véhicule correspondant utilisé. Ils ont ensuite formulé ce problème à l'aide d'un programme linéaire en nombres binaires. Ce dernier utilise deux types de variables : le premier détermine les arcs traversés par le véhicule et le second indique pour tout couple de sommets si le premier est traversé avant le second dans le circuit du véhicule. Le nombre de variables et de contraintes est polynomial. La relaxation linéaire est renforcée par l'introduction de contraintes valides et le problème est résolu par un algorithme de coupes et branchements.

2.3.1.2 PRL avec contraintes de fenêtres de temps

Certains travaux de recherche sur le problème de ramassage et livraison prennent en compte les contraintes de fenêtres de temps. Pour chaque demande, deux intervalles, appelées *fenêtres de temps*, sont respectivement associés à l'origine et la destination de la demande. Une durée est également associée à chaque arête du graphe (généralement identifiée au coût de l'arc). Les contraintes de fenêtres de temps limitent alors les trajets des véhicules à ceux pour lesquels le chargement et le déchargement des demandes

s'effectuent durant les intervalles de temps associés à leurs origines et destinations. Le problème est alors appelé *Problème de Ramassage et Livraison avec Fenêtres de Temps (PRLFT)*. Si la flotte de véhicule est constituée d'un unique (respectivement de plusieurs) véhicule(s), on parle alors du PRLFT mono-véhicule (1-PRLFT) (respectivement PRLFT multivéhicules (m-PRLFT)). Ces contraintes de fenêtres de temps complexifient le problème en introduisant une composante temporelle mais diminuent considérablement le nombre de solutions réalisables (surtout si les intervalles sont petits). Ceci permet alors d'appliquer efficacement certains algorithmes, notamment ceux basés sur la programmation dynamique. Nous donnons maintenant un aperçu des principaux travaux relatifs au problème de ramassage et livraison avec contraintes de capacité et fenêtres de temps. Pour un état plus complet (ou pour le cas sans capacité), se référer aux travaux [16, 66, 76].

Plusieurs algorithmes exacts ont été développés pour résoudre le PRLFT. Psaraftis [79] a modifié son algorithme de programmation dynamique pour prendre en compte les fenêtres de temps (l'objectif consiste alors à minimiser le temps de trajet du véhicule). Plusieurs travaux proposent de résoudre le PRLFT à l'aide d'algorithmes de génération de colonnes et branchements (voir [12, 60, 63, 97, 99] pour une présentation de cette méthode). Le problème maître correspond alors à un problème de partitionnement des demandes (par rapport aux trajets des véhicules) puisque chaque demande doit être servie exactement une fois. Le sous-problème consiste à déterminer un trajet réalisable de coût minimum (par rapport aux variables duales). Dumas et al. [31] sont les premiers à appliquer cet algorithme. Ils résolvent le sous-problème à l'aide de la programmation dynamique. Savelsbergh et Sol [88] ont amélioré cette méthode en résolvant le sous-problème de manière heuristique (si aucune colonne intéressante n'est trouvée, le sous-problème est alors résolu de manière exacte par programmation dynamique). Ils appliquent également une heuristique primale à chaque sommet de l'arbre de branchements, introduisent de nouvelles règles de branchements et modifient la gestion du pool de variables. L'algorithme est ensuite étendu afin de prendre en compte des contraintes additionnelles et d'être utilisé dans un contexte dynamique. Des instances réelles (contenant des contraintes additionnelles) ont été résolues par Xu et al. [100]. Sigurd et al. [93] ont également appliqué cet algorithme dans le cas du transport d'animaux vivants. Ce problème se caractérise par des restrictions sur les trajets des véhicules liés à la propagation de virus. À chaque demande (*i.e.*, transport d'un cheptel) est associé un état de santé. Afin d'éviter toute contagion, aucun animal ne peut voyager dans un véhicule contenant ou ayant contenu un autre animal avec un état de santé plus dégradé. La résolution du sous-problème est alors modifiée pour prendre en compte de telles contraintes. Ropke et al. [83] ont amélioré l'algorithme de génération de colonnes et branchements par l'ajout d'inégalités valides pour le problème maître à

chaque nœud de l'arbre, conduisant la méthode à un algorithme de coupes, génération de colonnes et branchements. Des instances contenant jusqu'à 500 demandes ont alors pu être résolues de manière optimale.

Des algorithmes de coupes et branchements ont également été implémentés pour résoudre le PRLFT. Lu et Dessouky [62] ont étendu leur modèle pour satisfaire les contraintes de temps en ajoutant un nombre polynomial de variables et de contraintes. Ce modèle est résolu de la même manière que le modèle sans fenêtres de temps. Ropke et al. [83] ont implémenté deux algorithmes de coupes et branchements pour le m-PRLFT. Ces algorithmes dépendent de deux formulations en nombres entiers contenant un nombre polynomial de variables binaires et un nombre exponentiel de contraintes. La première formulation s'appuie sur les contraintes de précédences définies par Ruland et Rodin [85] qui assurent que l'origine de chaque demande est visitée avant la destination et que ces deux sommets sont visités par le même véhicule. La deuxième formulation s'appuie sur les contraintes de capacité arrondies [69] et les contraintes d'élimination de chemins irréalisables [8]. Pour les deux modèles, les auteurs renforcent la relaxation linéaire en ajoutant des inégalités valides.

Le problème de ramassage et livraison avec fenêtres de temps a également été considéré d'un point de vue heuristique. Savelsbergh [86] a proposé une implémentation efficace de la recherche d'une solution réalisable dans le voisinage d'une solution du PRLFT, lorsque ce voisinage est défini par l'opérateur k-inter-échange. (L'opérateur k-inter-échange consiste à supprimer dans un cycle hamiltonien k arêtes puis à en ajouter k nouvelles afin d'obtenir un autre cycle hamiltonien.) Van der Bruggen et al. [96] ont développé une heuristique à deux phases pour le PRLFT mono-véhicule. L'objectif consiste à minimiser la durée du trajet du véhicule. Les deux phases de l'heuristique correspondent à une recherche locale. Dans la première phase, l'objectif consiste à minimiser la violation des contraintes de fenêtres de temps afin d'obtenir une solution réalisable. La deuxième phase minimise la durée du trajet tout en maintenant la réalisabilité des solutions. Cette heuristique a été testée sur des instances contenant jusqu'à 50 demandes. Les résultats expérimentaux ont montré que les solutions obtenues sont proches de l'optimum. De plus, la qualité des solutions obtenues et la vitesse d'exécution de l'heuristique augmente lorsque les intervalles des fenêtres de temps diminue. Nanry et Barnes [70] ont développé une recherche tabou réactive (l'algorithme, introduit dans [13], ajuste automatiquement les stratégies et les paramètres de recherche à chaque itération) pour le PRLFT sans limitation sur le nombre de véhicules. La solution initiale est obtenue en insérant au coût minimum les demandes dans le trajet du véhicule courant de manière itérative (si aucune insertion n'est possible, un nouveau véhicule est utilisé). La recherche tabou utilise trois opérateurs de voisinage : insertion d'une demande dans un autre trajet ; échange de deux demandes appartenant à

des trajets différents et déplacement d'une origine ou d'une destination dans le trajet du véhicule. Des solutions irréalisables par rapport aux contraintes de capacité et de fenêtres de temps sont temporairement permises en contrepartie d'une pénalité dans la fonction objective. Lau et Liang [56] ont utilisé ces résultats pour résoudre le même problème, à la différence qu'un coût fixe d'utilisation d'un véhicule est maintenant considéré. L'heuristique de construction initiale est modifiée et leur recherche tabou (non réactive) utilise les trois mêmes opérateurs. Cependant, les auteurs ont introduit la notion de cluster (regroupement de demandes dont les origines et les destinations sont proches) et adapté ces opérateurs afin de pouvoir les utiliser avec les clusters. Bent et Van Hentenrick [15] ont également résolu le même problème à l'aide d'une heuristique en deux phases. La première correspond à un recuit simulé où le nombre de véhicules doit être minimisé. La deuxième phase consiste alors à minimiser la distance parcourue, le nombre maximum de véhicules disponibles étant fixé (et donné par la première phase). Pour cela, l'heuristique utilise un voisinage de recherche large [91] correspondant à l'ensemble des solutions pouvant être obtenues en réinsérant z demandes, où z est un paramètre de l'algorithme. La réinsertion s'effectue de manière approchée à l'aide d'un algorithme de séparation et évaluation tronqué. Ropke et Pisinger [84] ont modifié l'algorithme de Bent et Van Hentenrick en utilisant pour la suppression et pour l'insertion plusieurs heuristiques, chacune donnant un voisinage différent. (La méthode est appelée méthode de recherche adaptative à voisinage large.) À chaque itération, le choix de l'heuristique pour l'insertion et la suppression s'effectue suivant la performance historique de chacune.

Ioachim et al. [50] ont proposé une méthode approchée pour résoudre le PRLFT pour des contraintes de capacité multi-dimensionnelles (les demandes ont des volumes et les véhicules des capacités de trois types différents). Cette heuristique est un algorithme de type regroupement en premier, parcours en second (cluster-first, route-second). La première phase consiste à créer des mini-clusters (un mini-cluster est un petit ensemble de demandes pouvant être servies par un même véhicule. Le véhicule entre et sort du mini-cluster à vide). Cette phase est résolue à l'aide d'un algorithme de génération de colonnes dont le sous-problème est résolu par programmation dynamique. Ces mini-clusters sont ensuite reliés entre eux pour former les trajets des véhicules. Cette seconde phase correspond au problème du voyageur de commerce avec fenêtres de temps multi-véhicules (de petite taille), résolu lui aussi par génération de colonnes. Cette heuristique a été testée sur des instances contenant jusqu'à 2500 demandes. Les résultats obtenus ont montré que cette heuristique donne des solutions meilleures de l'ordre de 6 à 10% que lorsque les mini-clusters sont formés à l'aide d'une heuristique d'insertion parallèle.

2.3.1.3 Problèmes proches du PRL

Des problèmes proches du problème de ramassage et livraison ont également été étudiés. Parmi ceux-ci, il existe le *Problème de la Grue (PG)* correspondant au 1-PRL lorsque le véhicule ne transporte qu'une seule demande à la fois mais peut traverser plusieurs fois chaque arête du graphe. Ce problème a été introduit par Frederickson et al. [40]. Ces derniers ont montré que ce problème est NP-difficile par réduction du problème du voyageur de commerce. Ils ont ensuite donné une $\frac{9}{5}$-approximation. Atallah et Kosaraju [9] ont considéré le PG lorsque le graphe est une ligne ou un cercle. Ils ont prouvé que ce problème est polynomial dans ces graphes. Frederickson et Guan [39] ont étudié le problème de la grue lorsque le graphe est un arbre. Ils ont montré que dans ce cas, le problème est NP-complet et fourni plusieurs α-approximations. Guan [46] a étendu ces résultats de complexité en prouvant que le problème de ramassage et livraison mono-véhicule, lorsque le véhicule peut traverser plusieurs fois chaque arête, est NP-complet même si le graphe est un cercle ou une ligne.

Le problème de la grue peut être étendu au cas des demandes multi-origines et multi-destinations (problème multi-multi) (voir [16] pour une classification des problèmes de transport). Dans cette extension, appelée le *Problème de l'Échange (PE)*, des coûts symétriques sont généralement considérés et chaque sommet est l'origine d'une demande et la destination d'une autre (une demande fictive peut être considérée si nécessaire). Chaque demande possède plusieurs origines et destinations. Le volume disponible (respectivement requis) à chaque origine (respectivement destination) est d'une unité et chaque unité d'une demande, chargée à une origine, peut être déchargée à n'importe quelle destination de la demande. Un unique véhicule d'une capacité de transport égale à un est disponible pour transporter ces demandes. Bordenave et al. [19] ont donné certaines propriétés structurelles des solutions optimales du problème de l'échange. Ces propriétés ont ensuite été utilisées pour modéliser le problème à l'aide d'un programme linéaire en nombres entiers. Les auteurs ont renforcé la relaxation linéaire par l'introduction de contraintes valides basées sur les contraintes du problème du voyageur graphique et ont développé un algorithme de coupes et branchements. Ce dernier leur a permis de résoudre rapidement des instances contenant jusqu'à 200 sommets et 8 demandes.

Hernández-Pérez et Salazar-González ont considéré une variante du problème de l'échange contenant une seule demande. Par contre, le volume disponible (respectivement requis) à chaque origine (respectivement destination) est un entier positif quelconque et le véhicule a une capacité supérieure à un. De plus, le véhicule doit visiter chaque client exactement une fois. Les auteurs [47] ont utilisé la même formulation que celle donnée pour le PRL dans [49], à la différence que le modèle s'appuie sur une

formulation du problème du voyageur de commerce symétrique et la demande est représentée par un seul type de flot. Ce problème est reformulé par application du lemme de Farkas pour l'existence d'un flot réalisable pour les demandes, donnant une formulation ayant un nombre exponentiel de contraintes de coupes en plus des contraintes du voyageur de commerce. Le modèle est renforcé par l'introduction d'inégalités valides pour le problème et résolu par un algorithme de coupes et branchements. Les auteurs ont également fourni une heuristique basée sur l'algorithme de coupes et branchements pour résoudre le problème de manière approchée [48]. Plutôt que d'exécuter l'algorithme sur l'ensemble des solutions, ce dernier est appliqué pour déterminer la meilleure solution parmi celles différant d'au maximum k arêtes (k est un paramètre de l'heuristique) d'une solution courante donnée. La solution initiale est donnée par une heuristique construisant de manière gloutonne un circuit qui est par la suite amélioré ou rendu réalisable par application des opérateurs 2-opt et 3-opt. L'algorithme de coupes et branchements est ensuite exécuté (avec une limite de temps et une profondeur maximum dans l'arbre de branchements) pour cette solution initiale. La nouvelle solution obtenue sert alors de point de départ pour une nouvelle application de l'algorithme de coupes et branchements et ce tant qu'une solution de coût inférieur est trouvée. Par ailleurs, afin d'accélérer la méthode, ils s'assurent que l'algorithme n'explore pas une région déjà visitée en imposant que le nombre d'arêtes différentes, par rapport à chacune des solutions précédentes, soit supérieur ou égal à $k + 1$.

2.3.2 Rechargements, transbordements et préemption

Les solutions obtenues pour le problème de ramassage et livraison montrent que la capacité des véhicules est rarement utilisée entièrement, c'est-à-dire qu'il est peu fréquent que la somme des volumes des demandes transportées par un véhicule à un instant donné soit égale à la capacité de ce dernier. Une technique pour réduire les coûts consiste alors à tenter d'utiliser la capacité de transport disponible (*i.e.*, la différence entre la capacité et le volume des demandes transportées) en transportant une autre demande. Cependant, la plupart du temps, les véhicules n'ont pas suffisamment de capacité pour transporter entièrement une autre demande. Une variante intéressante du PRL, appelée *Problème de Ramassage et Livraison avec Rechargements (PRLR)*, consiste alors à permettre aux véhicules d'interrompre temporairement le transport des demandes en les déchargeant sur des sommets qui ne correspondent pas à leur destination. Cette possibilité offre deux avantages. Elle permet à un véhicule d'augmenter sa capacité de transport du volume des demandes déchargées et de transporter ainsi une ou plusieurs demande(s) avant de revenir recharger les demandes déchargées. De plus, il est possible d'effectuer des transbordements, c'est-à-dire un changement de véhicule

lors du transport d'une demande, si le véhicule qui décharge la demande est différent de celui qui la recharge. Dans les deux cas, le fait de décharger/recharger une demande est appelé un *rechargement*. Le terme *transbordement* est employé si les véhicules déchargeant et rechargeant une demande sont différents. Il est à noter qu'aucune contrainte ni aucun coût n'est considéré pour les rechargements.

Nous donnons maintenant un exemple afin de montrer le gain engendré par l'introduction des rechargements dans le transport des demandes. Dans cet exemple, donné par la figure 2.1, chaque arête représente deux arcs opposés et même si tous les arcs ne sont pas représentés, le graphe est supposé complet. Les nombres correspondent aux coûts associés aux arcs et l'on peut déduire le coût des arcs non représentés en calculant un plus court chemin entre leurs deux extrémités selon les coûts de la figure. Chaque demande $k \in \{1, 2, 3\}$ possède un volume égal à 1 et son origine o^k et sa destination d^k sont indiquées sur la figure. De plus, un seul véhicule de capacité 1 est disponible au dépôt v_0. On suppose que pour la version avec rechargements, la demande 2 peut être déchargée/rechargée au sommet v_1.

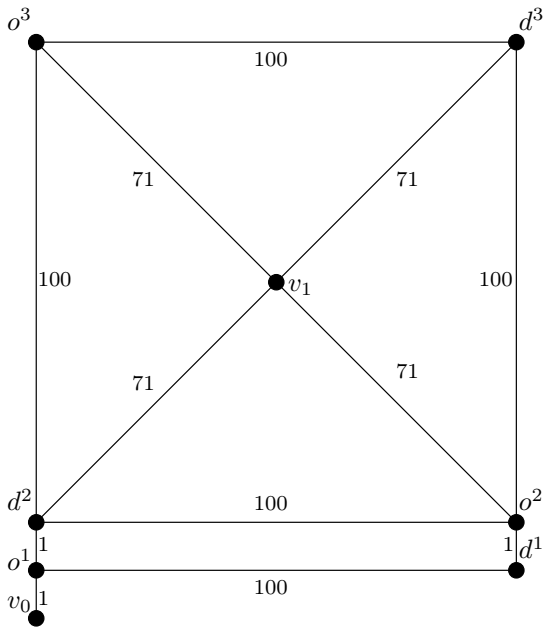

FIGURE 2.1 – Intérêt des rechargements dans le transport des demandes

Une solution optimale du PRL est la suivante. Le véhicule part du dépôt v_0 pour transporter la demande 1 de son origine o^1 à sa destination d^1. Il va ensuite à l'origine de la deuxième demande et la transporte jusqu'à sa destination d^2. Le véhicule se dirige ensuite en o^3 et transporte la demande 3 jusqu'à sa destination d^3 puis retourne au dépôt. Le coût associé est de 546. Lorsque le véhicule peut transporter les demandes

avec rechargements, on obtient une solution de coût inférieur de la manière suivante. Le véhicule part du dépôt et transporte la demande 1 de son origine à sa destination. Il va ensuite en o^2, charge la demande 2 puis la transporte jusqu'au sommet v_1 où il décharge cette demande. Ce faisant, le véhicule a maintenant suffisamment de capacité pour transporter la demande 3. Il va donc au sommet o^3 où il charge la demande 3 et la transporte jusqu'à sa destination d^3. Il retourne ensuite au sommet v_1 pour recharger la demande 2 et terminer son transport à sa destination d^2. Le véhicule termine en retournant au dépôt. Le coût de cette solution est alors de 488.

Mitrović-Minić et Laporte [67] ont évalué expérimentalement l'intérêt des transbordements dans le transport des demandes pour le problème de ramassage et livraison. Ils ont montré, de manière empirique, que les transbordements permettent d'obtenir des gains pouvant aller jusqu'à 40%.

L'introduction des rechargements dans le transport des demandes change la structure du problème. Tout d'abord, même si l'on considère un graphe complet tel que chaque sommet (excepté les dépôts) est incident à au plus une demande et un vecteur coût satisfaisant les inégalités triangulaires, l'ensemble V ne peut être restreint à l'ensemble des dépôts et des origines et destinations des demandes. Les sommets incidents à aucune demande et ne correspondant à aucun dépôt peuvent en effet être utilisés pour effectuer des rechargements, comme le montre l'exemple de la figure 2.1. L'ensemble V doit donc contenir un sommet pour chaque nœud du réseau.

L'introduction des rechargements modifie aussi profondément la structure des solutions. En effet, lorsqu'une demande est déchargée puis rechargée sur un sommet par un même véhicule, celui-ci traverse alors deux fois ce sommet : une première fois pour décharger la demande et une seconde fois pour la recharger. Le circuit représentant le trajet d'un véhicule n'est alors plus élémentaire. De plus, dans le cas d'un transbordement ou si deux véhicules effectuent un rechargement sur un même sommet, les circuits représentant les trajets des véhicules ne sont plus sommet-disjoints. Finalement, les rechargements impliquent que les chemins des demandes ne sont plus des sous-chemins des circuits des véhicules. Dans les cas des rechargements ne correspondant pas à des transbordements, comme le véhicule continue son trajet pendant qu'une demande est déchargée temporairement sur un sommet, le trajet de la demande jusqu'à ce sommet et celui à partir de ce sommet ne sont pas consécutifs dans le circuit du véhicule. Dans le cas des transbordements, la demande est transportée par plusieurs véhicules donc son chemin ne peut être sous-chemin d'un unique circuit.

La "destructuration" des solutions, *i.e.*, le fait que les circuits ne sont plus élémentaires ni sommet-disjoints et les chemins des demandes ne sont plus des sous-chemins

des circuits des véhicules, entraîne que les conditions données dans la section 2.2 pour qu'une solution du PRL soit réalisable ne sont plus valides. Premièrement, la condition 2 forçant l'origine et la destination de chaque demande à être traversées par le même véhicule n'est plus vraie si les transbordements sont autorisés. De plus, la condition 3, imposant que l'origine d'une demande soit traversée avant sa destination n'a plus de sens puisqu'un sommet peut être traversé plusieurs fois par un même véhicule. Il est donc nécessaire de définir les conditions nécessaires et suffisantes pour qu'une solution du PRL avec rechargements soit réalisable.

Tout d'abord, puisque les chemins des demandes ne sont plus des sous-chemins des circuits des véhicules, nous définissons une solution du PRLR à l'aide des circuits des véhicules et des chemins des demandes (Nous montrerons dans le chapitre 4 qu'il est nécessaire de connaître les chemins des demandes pour déterminer en temps polynomial si une solution est réalisable). Sans perte de généralité, nous considérons que les trajets des demandes sont représentés par des chemins élémentaires. En effet, il est toujours possible de supprimer les circuits dans les chemins des demandes en remplaçant chaque circuit par un rechargement. (Ceci est possible puisqu'aucune contrainte ni au aucun coût n'est considéré pour les rechargements.) Une solution du PRLR est alors représentée par un ensemble de circuits $C_1, C_2, \ldots, C_{|F|}$ correspondant aux circuits des véhicules et un ensemble de chemins élémentaires $L_1, L_2, \ldots, L_p$ correspondant aux trajets des demandes. Cette solution est réalisable pour le problème de ramassage et livraison avec rechargements si la solution satisfait les conditions suivantes (les conditions 5 et 6 sont les mêmes que les conditions 1 et 4 données dans la section 2.2) :

5 - le cycle C_f passe par w_f pour tout $f \in F$,

6 - la quantité transportée par le véhicule $f \in F$ ne dépasse pas Q_f,

7 - L_k, $k \in K$, est un chemin élémentaire de o^k à d^k,

8 - les chemins des demandes n'utilisent que des arcs traversés par les véhicules,

9 - il existe un ordre sur les arcs des circuits des véhicules *conforme* aux séquences des circuits des véhicules et des chemins des demandes.

La dernière condition s'explique par le fait que tout chemin ou circuit correspond à une séquence (*i.e.*, suite ordonnée) d'arcs. Elle signifie qu'il existe un ordre sur les arcs des circuits des véhicules tel qu'un arc a est traversé avant un arc a' si et seulement s'il n'existe aucune demande ou véhicule traversant l'arc a' avant l'arc a. Si les transbordements ne sont pas permis, il faut renforcer la condition 6 en stipulant que chaque chemin associé à une demande utilise uniquement les arcs traversés par un seul véhicule.

Les conditions pour qu'une solution soit réalisable sont donc complexifiées par l'introduction des rechargements. Trouver une description des solutions du problème per-

mettant de savoir en temps polynomial si une solution est réalisable est donc un point essentiel de la modélisation et sera donc largement étudié dans cette thèse, notamment dans le chapitre 4.

Nous présentons maintenant les deux versions du problème de ramassage et livraison avec rechargements étudiées dans la littérature. Dans la première, appelée *Problème de Ramassage et Livraison avec Transbordements (PRLT)*, les rechargements peuvent correspondre à des transbordements (le nombre de véhicules disponibles est strictement supérieur à un) et ne peuvent se faire que sur certains sommets, appelés *sommets de transbordements*. Les travaux concernant cette version considèrent également les contraintes de fenêtres de temps. La deuxième version étudiée est appelée *Problème de Ramassage et Livraison Préemptif mono-véhicule (1-PRLP)*. Comme seul un véhicule est disponible, il est évident que les rechargements ne sont pas des transbordements. Par contre, le terme préemptif fait référence au fait que les rechargements peuvent se faire sur tous les sommets. Nous discutons d'abord de certains travaux portant sur le PRLT puis considérons le 1-PRLP et différentes variantes.

Dans [74], Oertel a proposé une heuristique pour résoudre le problème de ramassage et livraison avec transbordements. Cette heuristique consiste à créer d'abord un graphe auxiliaire orienté en considérant plusieurs copies pour chaque sommet de transbordements. En fait, chacun de ces derniers est divisé en deux sommets pour chaque demande. Cette transformation assure que le circuit de chaque véhicule est élémentaire et que tous ces circuits sont sommet-disjoints. Sur ce nouveau graphe, le problème est modélisé à l'aide d'un programme linéaire mixte. Le problème est finalement résolu sur le graphe auxiliaire à l'aide d'une recherche tabou dans laquelle les solutions voisines sont déterminées par suppression puis réinsertion (avec ou sans transbordement) d'une demande. Des instances ayant environ 70 demandes et un sommet de transbordements ont été résolues en utilisant cette heuristique.

Mitrović-Minić et Laporte [67] ont donné une heuristique en deux phases pour résoudre de manière approchée le PRLT sans contraintes de capacité. (Ce travail est la seule référence du problème de ramassage et livraison sans contrainte de capacité que nous donnons. Ceci s'explique par le fait que les transbordements apparaissent même sans limitation de la capacité du véhicule.) Nous remarquons que dans ce cas, tous les rechargements sont des transbordements. Les auteurs ont d'abord construit une solution initiale en utilisant une procédure d'insertion de coût minimum avec plusieurs points de départ. La meilleure solution trouvée est utilisée comme solution initiale. Cette dernière est alors améliorée en supprimant et insérant successivement chaque demande. Dans les deux phases de l'heuristique, une demande peut être insérée avec un transbordement au maximum. Ce choix est fait en considérant toutes les insertions

possibles et en choisissant la meilleure. L'insertion d'une demande k, transportée de o^k à d^k avec un rechargement au sommet v, sera représentée par deux demandes ayant respectivement o^k et v comme origines et v et d^k comme destinations. Les fenêtres de temps des deux demandes sont choisies pour s'assurer que k est transportée dans ses fenêtres de temps et que le premier segment de la demande (correspondant au chemin allant de o^k à v) est transporté avant le deuxième (de v à d^k). Comme mentionné plus haut, les résultats expérimentaux qu'ils ont obtenus montrent que permettre les transbordements est intéressant pour réduire les coûts des solutions.

Cortés et al. [25] ont considéré le PRLT dans lequel seuls les transbordements sont autorisés (mais pas les rechargements avec un seul véhicule). Ils ont construit un graphe auxiliaire en créant plusieurs copies pour chaque sommet de transbordements. Le nombre de copies est égal à deux fois le nombre maximum de fois qu'un véhicule peut effectuer de transbordements en un sommet (cette limite est fixée arbitrairement). À l'aide de ce graphe, ils ont donné un programme linéaire mixte pour le problème s'appuyant sur une approche arc-sommets. Le problème est ensuite résolu par un algorithme basé sur la décomposition de Benders. Cette méthode a été appliquée pour des instances contenant jusqu'à 6 demandes, 2 véhicules et 1 sommet de transbordements.

Peu de travaux portent sur le problème de ramassage et livraison préemptif mono-véhicule. À notre connaissance, seuls des résultats de complexité existent pour ce problème lorsque le véhicule peut traverser plusieurs fois chaque arête. Guan [46] a montré que ce problème est polynomial si le graphe est une ligne ou un cercle (il fournit un algorithme de même complexité que celui donné dans le cas où le véhicule ne transporte qu'une seule demande à la fois [9]) mais NP-difficile si le graphe est un arbre.

Le *Problème de la Grue Préemptif*, correspondant au 1-PRLP dans lequel le véhicule peut transporter une seule demande à la fois a été introduit par Atallah et Kosaraju [9]. Ces derniers ont considéré ce problème lorsque le graphe est une ligne ou un cercle. Ils ont prouvé que dans ce cas, le problème peut être résolu en temps polynomial. Frederickson et Guan [38] ont étendu cette étude en montrant que le problème reste polynomial si le graphe est un arbre et ont donné deux algorithmes combinatoires exacts.

Le *Problème de l'Échange Préemptif (PEP)* a également été considéré, ainsi que le *Problème de l'Échange Mixte (PEM)* dans lequel une partie des demandes peut être transportée avec préemption et l'autre sans aucun rechargements. Anily et Hassin [3] ont donné une $\frac{5}{2}$-approximation pour la version mixte du problème. Dans [2], ils ont considéré le problème de l'échange préemptif lorsque le graphe est un arbre. Ils ont prouvé que le problème reste NP-complet et ont donné pour ce cas une $\frac{3}{2}$-approximation.

Ils ont aussi montré que le problème est polynomial s'il n'y a qu'une seule demande. Récemment, Bordenave et al. [20] ont présenté un algorithme de coupes et branchements pour résoudre le problème de l'échange préemptif qui leur a permis de résoudre des instances comportant jusqu'à 100 sommets et 8 demandes. Ils ont également considéré une heuristique en deux phases [21] pour le problème de l'échange mixte asymétrique. La phase de construction est basée sur le travail réalisé par Anily et Hassin [3]. La solution obtenue est ensuite améliorée en appliquant des échanges d'arcs et en modifiant les circuits où aucune demande n'est transportée avec rechargements. Afin de mesurer expérimentalement la qualité de l'heuristique, une borne inférieure est calculée en résolvant, pour chaque demande, le problème de l'affectation dans le graphe biparti orienté formé par les origines et les destinations de la demande. La borne est égale à la somme des coûts des solutions obtenues pour toutes les demandes. L'heuristique a été appliquée sur des instances contenant jusqu'à 10 000 sommets. L'écart moyen entre les solutions fournies par l'heuristique et les bornes inférieures ne dépasse pas 1%.

2.3.3 Fractionnement de la demande

Une autre façon d'améliorer les solutions du problème de ramassage et livraison consiste à permettre aux véhicules de transporter seulement une fraction (ou partie) de chaque demande. Dans cette variante du PRL, appelée *Problème de Ramassage et Livraison avec Fractionnement (PRLF)*, les demandes peuvent alors être fractionnées en plusieurs parties, la somme des parties devant correspondre au volume de la demande. Chaque partie peut alors être transportée par n'importe quel véhicule indépendamment des autres. On suppose que chaque partie correspond à la quantité chargée dans un véhicule lors d'un de ses passages. Un même véhicule peut effectuer plusieurs chargements, *i.e.*, passer plusieurs fois par l'origine de la demande et charger une partie de cette demande à chaque passage. Comme pour le PRL, lorsqu'une demande ou partie de la demande est chargée dans le véhicule, elle ne peut être déchargée que sur sa destination (*i.e.*, les rechargements sont interdits). La variante du PRL dans laquelle les véhicules peuvent également interrompre momentanément le transport d'une demande sera étudiée dans la section suivante.

Il est évident que le fractionnement des demandes pour le problème de ramassage et livraison permet dans certains cas d'obtenir de meilleures solutions puisque le PRLF est une relaxation du PRL. Nous donnons ici un exemple dans lequel le coût de la solution optimale du PRLF est strictement inférieur au coût de la solution optimale du PRL. Dans cet exemple, nous avons un seul véhicule chargé de transporter trois demandes. Considérons le graphe donné dans la figure 2.2. Comme pour le graphe donné dans la

figure 2.1, chaque arête représente deux arcs opposés et même si tous les arcs ne sont pas représentés, le graphe est considéré complet. Les nombres correspondent aux coûts associés aux arcs et nous pouvons déduire le coût des arcs non représentés en calculant un plus court chemin entre leurs deux extrémités selon les coûts de la figure. Chaque demande $k \in \{1, 2, 3\}$ possède un volume égal à 2 et son origine o^k et sa destination d^k sont indiquées sur la figure. De plus, un seul véhicule de capacité 3 est disponible au dépôt v_0.

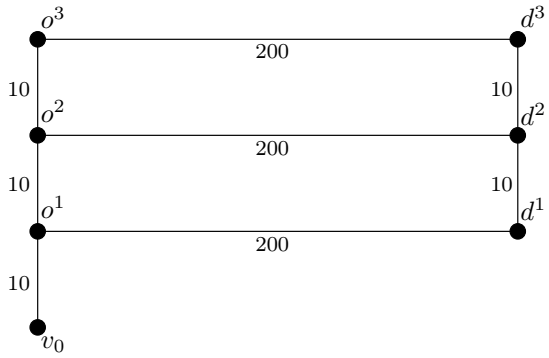

FIGURE 2.2 – Intérêt du fractionnement des demandes

Dans une solution optimale du problème de ramassage et livraison, le véhicule part du dépôt v_0 pour aller au sommet o^1 où il charge la demande 1 qu'il transporte jusqu'à sa destination d^1. Le véhicule arrive ensuite au sommet o^2 pour transporter la demande 2 jusqu'à sa destination d^2. Il va ensuite à l'origine de la demande 3, la charge et la transporte directement jusqu'à son origine d^3 puis retourne au dépôt. Le coût associé est de 1260. Pour le problème de ramassage et livraison avec fractionnement, nous obtenons une solution de coût inférieur. Le véhicule sort du dépôt v_0 et arrive au sommet o^1 pour charger la demande 1. Il va ensuite au sommet o^2 pour charger une unité de la demande 2. À ce moment-là, le véhicule est complètement chargé. Il va ensuite au sommet d^2 pour décharger la fraction de la demande 2 qu'il transportait puis en d^1 pour décharger la demande 1. Le véhicule retourne à vide à l'origine de la demande 2 pour terminer son chargement. À ce moment-là, sa capacité disponible est de 2 et le véhicule peut alors transporter la demande 3. Il va donc au sommet o^3 pour charger la demande puis la transporte jusqu'à sa destination d^3. Il retourne ensuite au sommet d^2 pour terminer le transport de la demande 2. Le véhicule retourne finalement au dépôt. Le coût associé à cette solution est de 880.

Bien que dans l'exemple précédent, le volume de chaque demande soit inférieur ou égal à la capacité du véhicule, nous pouvons remarquer que cette condition n'est plus nécessaire pour qu'une instance du problème admette une solution réalisable.

S'il est certain que le fractionnement des demandes permet d'obtenir des solutions de meilleur coût, il est également évident que cette possibilité complexifie considérablement le problème. En effet, chaque demande est maintenant transportée sur plusieurs chaînes. De plus, le cycle de chaque véhicule n'est plus élémentaire et les cycles des véhicules ne sont plus sommet-disjoints. (Cependant, contrairement au PRLR, les chaînes des demandes sont des sous-chaînes des cycles des véhicules).

Toute solution du PRLF consiste en un cycle associé à chaque véhicule et en un ensemble de chaînes associé à chaque demande, puisque les demandes peuvent être transportées sur plusieurs chemins. La quantité de demande transportée sur chaque chaîne élémentaire est également spécifiée. Les conditions pour qu'une solution du PRLF soit réalisable sont donc les mêmes que pour le problème de ramassage et livraison avec rechargements, à la différence que chaque chaîne d'une demande doit être une sous-chaîne d'un cycle d'un véhicule puisque les rechargements sont interdits. De plus, il faut s'assurer que pour toute demande, la somme des quantités transportées sur les chaînes élémentaires associées à cette demande corresponde au volume de celle-ci. Il est donc évident que les modèles et les algorithmes donnés pour le problème de ramassage et livraison ne peuvent être facilement étendus pour prendre en compte cette possibilité.

Malgré l'intérêt du problème de ramassage et livraison avec fractionnement, ce dernier n'a quasiment jamais été traité dans la littérature. À notre connaissance, seuls Nowak et al. [73] ont étudié ce problème. Ces derniers ont initialement considéré le cas mono-véhicule et supposé que les volumes des demandes peuvent être strictement supérieurs à la capacité du véhicule. De plus, chaque sommet peut être incident à plusieurs demandes. Ils ont tout d'abord montré que le gain entre les coûts des solutions du PRLF et du PRL est maximisé lorsque le volume de chaque demande est juste supérieur à la moitié de la capacité du véhicule. Ils ont ensuite résolu le problème à l'aide d'une métaheuristique. À chaque itération, la solution courante est modifiée en fractionnant le transport d'une demande (une liste tabou est utilisée pour sauvegarder les différents fractionnements effectués). Cette nouvelle solution est ensuite améliorée à l'aide d'opérateurs de voisinage définis pour le PRL (*i.e.*, sans fractionnement) et correspond alors à la solution voisine. Si cette dernière a un coût inférieur, la solution courante est mise à jour. Initialement, la solution courante est donnée par une heuristique résolvant le problème sans fractionnement. Les résultats expérimentaux sur des instances aléatoires montrent que prendre en compte le fractionnement des demandes permet d'obtenir des gains allant jusqu'à 40%. Ils ont ensuite modifié leur algorithme pour prendre en compte le cas multi-véhicules avec des coûts associés à chaque chargement et chaque déchargement et des coûts fixes d'utilisation de véhicules. Les résultats obtenus montrent que dans ce cas, le gain en termes de coût est négligeable par rapport au problème de ramassage et livraison. Les solutions du PRLF utilisent cependant

moins de véhicules.

Si le fractionnement des demandes a été très peu étudié pour le problème de ramassage et livraison, il a cependant fait l'objet de plusieurs travaux de recherche pour le *Problème de Tournées de Véhicules (PTV)*. Ce dernier (dans sa version sans fractionnement) peut être décrit comme suit. Soit I un ensemble de clients. Une quantité d_i, correspondant au volume, doit être acheminée pour chaque client $i \in I$ à partir d'un dépôt central à l'aide d'une flotte homogène de véhicules (la capacité des véhicules est supérieure ou égale au volume requis par chaque client). Les coûts entre deux clients et entre chaque client et le dépôt sont symétriques et satisfont les inégalités triangulaires. La livraison des clients doit se faire selon un coût total minimum correspondant à la somme des coûts de chaque tournée de véhicule, tout en s'assurant que le volume total des clients desservis par chaque véhicule est inférieur ou égal à la capacité du véhicule et chaque client est servi par un seul véhicule. Le PTV peut être vu comme un cas particulier du problème de ramassage et livraison dans lequel les origines des demandes correspondent au dépôt. Ce problème, introduit par Dantzig et Ramser [27], a été largement étudié depuis plusieurs décennies. Le *Problème de Tournées de Véhicules avec Fractionnement (PTVF)* consiste à permettre à chaque client d'être desservi par plusieurs véhicules, chacun délivrant une fraction du volume du client. Dans ce problème, nous remarquons que les volumes des clients peuvent maintenant être supérieurs à la capacité des véhicules. Nous présentons ici quelques travaux de recherche relatifs au PTVF. Pour un état de l'art plus complet sur le PTVF, le lecteur peut se référer à [5].

Le problème de tournées de véhicules avec fractionnement a été introduit par Dror et Trudeau [29, 30]. Dans ce travail, les auteurs ont développé une métaheuristique pour résoudre ce problème lorsque la flotte est de taille infinie et le volume de chaque client est inférieur ou égal à la capacité des véhicules. L'heuristique consiste à améliorer une solution initiale, correspondant à une solution du PTV trouvée heuristiquement, à l'aide de quatre opérateurs de voisinage. Deux d'entre eux sont des opérateurs de voisinage utilisés dans le PTV (sans fractionnement). L'un consiste à supprimer un client d'une route d'un véhicule pour l'insérer dans une autre ou à échanger deux clients appartenant à deux routes différentes. L'autre opérateur est le 2-inter-échange, qui revient à supprimer deux arêtes dans un circuit (correspondant au trajet d'un véhicule) et à former un autre circuit en reliant les deux parties déconnectées à l'aide de deux nouvelles arêtes. Les deux autres opérateurs sont spécifiques au fractionnement et fonctionnent comme suit. Le premier choisit un client i et un nombre j de véhicules tels que la somme des capacités de transport disponibles pour les j véhicules soit supérieure ou égale à la quantité d_i requise par le client. La solution voisine est alors obtenue en supprimant le client i dans toutes les routes le desservant et en l'insérant au meilleur coût dans chacune des routes des j véhicules. Le second consiste à diminuer

le nombre de clients servis avec fractionnement en ajoutant un nouveau véhicule. Pour cela, il choisit un client i servi par plusieurs véhicules et supprime i dans chacune des routes associées. Le client i est alors servi par un nouveau véhicule partant du dépôt, le servant et revenant directement au dépôt. L'heuristique consiste à appliquer les quatre opérateurs de voisinage jusqu'à ne plus pouvoir obtenir une amélioration de la solution courante. Les auteurs ont ensuite étudié les résultats obtenus pour le PTV et PTVF sur des instances générées aléatoirement contenant entre 75 et 150 clients. Les résultats expérimentaux montrent que si les volumes des clients ne sont pas trop faibles par rapport à la capacité des véhicules, la prise en compte du fractionnement dans le transport de la demande permet d'obtenir des gains atteignant jusqu'à 14% sur le coût total et jusqu'à 25% sur le nombre de véhicules utilisés. Archetti et al. [6] ont développé une recherche tabou pour ce problème. La solution initiale est obtenue de la manière suivante. Tous les clients ayant un volume supérieur à la capacité des véhicules sont servis par des véhicules effectuant des trajets directs entre le dépôt et le client, jusqu'à ce que le volume de chaque client soit strictement inférieur à la capacité (les clients avec un volume nul sont alors supprimés). Un problème de voyageur de commerce est résolu à l'aide de l'heuristique GENIUS [43] qui construit un cycle hamiltonien initial à l'aide d'une procédure d'insertion de sommets puis l'améliore par suppression/réinsertion d'un sommet dans le cycle. La solution du problème du voyageur de commerce obtenue est alors découpée en plusieurs tournées afin d'obtenir des tournées réalisables par rapport à la capacité. Les solutions voisines sont déterminées par rapport à la solution courante par insertion de coût minimum d'un client dans une route ne le servant pas. Afin de permettre le fractionnement de la demande, l'insertion d'un client peut se faire sans que ce dernier soit supprimé des routes le servant. Une postoptimisation de chaque tournée est effectuée avec l'algorithme GENIUS. Les solutions obtenues par cette heuristique sont meilleures que celles obtenues dans [29, 30] mais cet algorithme nécessite plus de temps de calcul. Cette heuristique a ensuite été améliorée par Archetti et al. [7] qui ont formulé une version contrainte du PTVF à l'aide d'un programme linéaire mixte basé sur une formulation arc-chemins. En effet, cette formulation ne modélise pas le PTVF puisque quelques solutions du PTVF ne vérifient pas certaines contraintes de la formulation. Cependant, la prise en compte de ces dernières permet d'accélérer la résolution de la formulation. Leur heuristique consiste alors à résoudre la formulation avec un algorithme de coupes, génération de colonnes et branchements, mais sur un sous-ensemble de variables. Ils ne considèrent en effet qu'un sous-ensemble de routes et seule une partie des clients peut être desservie par plusieurs véhicules (*i.e.*, avec fractionnement). Le problème est donc résolu de manière exacte sur différents sous-ensembles de taille réduite, ces derniers étant choisis à partir des différentes solutions trouvées par la méthode tabou [6]. Cette heuristique leur a permis d'améliorer dans quasiment tous les cas les solutions obtenues par la méthode tabou.

De nombreux algorithmes permettant de calculer des bornes sur les coûts des solutions optimales du PTVF ont été mises en œuvre. Dror et al. [28] ont donné une formulation linéaire mixte pour le PTVF (sans limite sur le nombre de véhicules disponibles) utilisant deux types de variables codant respectivement les arcs traversés par les véhicules et la proportion de chaque demande transportée par chaque véhicule. Ils ont ensuite introduit plusieurs familles de contraintes valides pour renforcer la relaxation linéaire du problème. La résolution par un algorithme de coupes de la relaxation linéaire de cette formulation fournit une borne inférieure pour le PTVF. La borne supérieure est fournie par l'heuristique définie précédemment [29]. Les résultats expérimentaux montrent que l'écart entre la borne supérieure et la borne inférieure ne dépasse pas 9%.

Belenguer et al. [14] ont calculé une borne inférieure pour le PTVF en considérant une formulation relaxée du problème. Cette formulation entière est définie par des contraintes de coupes. Si toute solution du PTVF vérifie les contraintes de cette formulation relaxée, l'inverse n'est pas vrai et il est même NP-complet de déterminer si une solution entière de la formulation relaxée correspond à une solution réalisable pour le PTVF. Cependant, dans la pratique, il est généralement facile de répondre à cette question. Les auteurs ont introduit des contraintes valides pour la formulation relaxée et effectué une étude polyédrale de cette dernière. Cette formulation a ensuite été résolue par un algorithme de coupes et branchements permettant d'obtenir une borne inférieure pour le PTVF. La borne supérieure est donnée par une métaheuristique introduite par Compos et Mota [22] pour le PTV ainsi que par la transformation de toute solution entière de la formulation relaxée en une solution du PTVF lorsque cela est possible.

Jin et al. [52] ont considéré le PTVF lorsque les volumes des demandes peuvent dépasser la capacité des véhicules et la taille de la flotte est égale au nombre minimum de véhicules nécessaires pour que l'instance admette une solution réalisable. Ils donnent dans ce cas des bornes inférieure et supérieure déterminées par un algorithme basé sur la génération de colonnes. Pour cela, ils ont donné une formulation entière du problème comportant un nombre exponentiel de variables. Chaque variable est associée à une route définie par un circuit et un volume associé à chaque client de ce circuit. La borne inférieure est alors donnée par la relaxation linéaire résolue par génération de colonnes. Le sous-problème, similaire à un problème de voyageur de commerce avec profits, est résolu de manière exacte par un algorithme de recherche avec bornes. Pour calculer la borne supérieure, les auteurs ont fixé la valeur d'une variable fractionnaire puis recalculé la relaxation linéaire. Ils ont réitéré ce processus jusqu'à ce que presque toutes les variables soient fixées. Le problème résultant est alors résolu de manière exacte par un algorithme de génération de colonnes et branchements. Leur algorithme permet d'améliorer les bornes données dans la littérature pour la plupart des instances

testées, notamment lorsque le volume des demandes est grand par rapport à la capacité des véhicules.

Récemment, les bornes inférieures connues ont été améliorées par Moreno et al. [68]. Ces derniers ont modélisé le PTVF asymétrique à l'aide d'une formulation entière basée sur un graphe auxiliaire. Dans ce dernier, chaque arc (i, j) est remplacé par plusieurs arcs $(i, j)^{qd}$, représentant le fait de partir du client i avec une quantité q pour délivrer d unités au client j. La construction de ce graphe auxiliaire se fait en temps pseudo-polynomial puisque le nombre d'arcs dépend de la capacité des véhicules et des volumes des demandes. La formulation est un programme linéaire en nombres entiers utilisant des variables indexées sur les arcs du graphe auxiliaire et des contraintes de flot. Les auteurs ont ensuite reformulé ce programme linéaire en nombres entiers à l'aide de la décomposition de Dantzig-Wolfe, obtenant ainsi une formulation avec un nombre exponentiel de variables. Le sous-problème consiste en un plus court chemin dans le graphe auxiliaire et peut être alors résolu en temps pseudo-polynomial. Les auteurs ont renforcé la valeur de la relaxation linéaire par l'introduction de contraintes valides en nombre exponentiel. Ils ont résolu la relaxation par un algorithme de coupes et génération de colonnes, permettant d'obtenir l'amélioration des bornes inférieures connues pour le PTVF.

De nouveaux résultats de complexité pour le PTVF ont été donnés par Archetti et al. [4]. Ces derniers ont montré que lorsque les coûts sont symétriques et satisfont les inégalités triangulaires, le PTVF avec un nombre infini de véhicules dont la capacité est égale à un ou deux peut être résolu en temps polynomial alors que ce problème est NP-complet lorsque la capacité des véhicules dépasse deux.

2.3.4 Rechargements et fractionnement

Nous venons de voir dans les deux sections précédentes que le transport des demandes avec rechargements ou fractionnement permettait d'obtenir des solutions de meilleur coût pour le problème de ramassage et livraison. La première possibilité permet aux véhicules de décharger temporairement une demande sur un sommet différent de sa destination. Il est également possible dans le cas de plusieurs véhicules de transporter les demandes avec transbordements. La deuxième laisse la possibilité à chaque véhicule de ne transporter qu'une partie d'une demande. Ces deux possibilités peuvent évidemment être combinées ensemble pour obtenir de meilleures solutions en termes de coût. Ce problème est appelé *Problème de Ramassage et Livraison avec Rechargements et Fractionnement (PRLRF)*. Les demandes peuvent alors être transportées à la fois avec

rechargements et fractionnement. De plus, lors d'un rechargement, un véhicule peut décharger seulement une fraction de la demande (contrairement au cas avec rechargements mais sans fractionnement dans lequel une demande doit être totalement déchargée). Il est évident que cette version du problème de ramassage et livraison permet de diminuer les coûts induits par les véhicules. Cependant, ce problème est beaucoup plus complexe que la version classique du problème puisqu'il combine à la fois les difficultés liées aux rechargements et celles liées au fractionnement.

Une solution du PRLRF est décrite avec la même information qu'une solution du PRLF, à savoir un ensemble de cycles correspondant aux trajets des véhicules et un ensemble de chaînes élémentaires (avec la quantité de la demande transportée sur chaque chaîne) pour chaque demande. De plus, les conditions nécessaires et suffisantes pour qu'une solution du PRLRF soit réalisable sont les mêmes que pour le PRLF, mis à part que les chaînes des demandes ne doivent plus être des sous-chaînes des cycles des véhicules.

Le PRLRF se décline en plusieurs cas, suivant que les rechargements peuvent être des transbordements ou non et suivant s'ils peuvent être effectués n'importe où sur le réseau ou seulement en certains sommets. La complexité de ce problème explique le fait que ce dernier soit très peu étudié. À notre connaissance, seule une variante du problème de ramassage et livraison avec transbordements en certains sommets et fractionnement de la demande a été considérée par Grünert et Sebastian [45]. Cette variante apparaît dans le problème du transport du courrier de la Poste Allemande AG. Dans ce dernier, le courrier est d'abord collecté auprès des clients et dans des boîtes aux lettres pour être acheminé au centre de tri (CT) le plus proche. Après avoir été trié, le courrier est envoyé de son CT d'origine à son CT de destination. Ce transport est effectué par véhicule, train ou avion. Finalement, après une dernière étape de tri, le courrier est délivré par des postiers du centre de tri de destination à l'adresse indiquée. Ce problème contient plusieurs problèmes d'optimisation. En effet, la collecte et la distribution du courrier (première et dernière étapes) correspondent respectivement aux problèmes de tournées de véhicules [27] et du postier chinois [35]. Le problème du transport du courrier à l'aide des véhicules entre les différents centres de tri est appelé *Problème de Conception de Réseaux avec Flots pour les Véhicules et les Requêtes (PCRFVR)*. Ce problème est similaire au problème de ramassage et livraison avec transbordements en certains sommets et fractionnement de la demande lorsqu'une flotte hétérogène est disponible. Il existe cependant quelques différences. Dans le PCRFVR, chaque fois qu'un véhicule passe par un sommet, il est automatiquement déchargé. Le courrier déchargé est alors indisponible pour une certaine période correspondant au temps nécessaire pour trier ce courrier. Par ailleurs, un coût associé au tri et au stockage du courrier à chaque sommet est considéré. Finalement, les coûts des trajets des véhicules sont proportionnels à la

quantité de courrier qu'ils transportent et des contraintes de fenêtres de temps fortes (*i.e.*, les intervalles sont courts) sont prises en comptes. Grünert et Sebastian [45] ont considéré un graphe auxiliaire basé sur des périodes de temps. Ils ont formulé le problème à l'aide d'un programme linéaire mixte mais n'ont pas proposé d'algorithme pour le résoudre.

Dans cette thèse, nous considérons d'abord le problème de ramassage et livraison muti-véhicules préemptif avec fractionnement lorsque les coûts sont asymétriques (chapitre 3). Nous nous intéressons ensuite au problème de ramassage et livraison mono-véhicule préemptif asymétrique, correspondant à une version restreinte du problème précédent lorsque le fractionnement de la demande est interdit et que la flotte n'est composée que d'un seul véhicule (chapitres 4-6).

Chapitre 3

Le problème de ramassage et livraison multi-véhicules asymétrique préemptif avec fractionnement

Dans ce chapitre, nous nous intéressons au Problème de Ramassage et Livraison multi-véhicules Asymétrique Préemptif avec Fractionnement (m-PRLAPF). Celui-ci est une variante du problème de ramassage et livraison multi-véhicules asymétrique dans laquelle les demandes peuvent être transportées avec préemption et fractionnement. Comme expliqué dans le chapitre 2, le terme fractionnement signifie que chaque demande peut être divisée en plusieurs parties, chaque partie pouvant être transportée sur un chemin différent et par un véhicule différent. La seule condition requise est que la somme des quantités de la demande transportées sur l'ensemble des chemins soit égale au volume de la demande. Le terme préemption signifie que les véhicules peuvent décharger temporairement l'intégralité ou une partie - puisque le fractionnement des demandes est possible - d'une (ou plusieurs) demande(s) qu'ils transportent en n'importe quel sommet du graphe. De plus, les demandes peuvent être transportées avec transbordements dans le cas où le véhicule déchargeant une demande (ou partie d'une demande) et le véhicule rechargeant cette demande (ou partie) sont différents. Aucune contrainte sur la quantité déchargée, ni sur le nombre de rechargements dans le transport d'une demande n'est considérée. De même, nous supposons que les coûts associés aux rechargements sont négligeables. Dans le chapitre précédent, nous avons donné deux exemples montrant l'intérêt de transporter les demandes avec rechargements et fractionnement. Nous avons également donné des conditions nécessaires et suffisantes pour qu'une solution du problème soit réalisable et à ce titre, avons montré la complexité liée à l'introduction des rechargements et du fractionnement dans le transport

des demandes.

Nous définissons maintenant clairement les hypothèses retenues pour le problème de ramassage et livraison multi-véhicules asymétrique préemptif avec fractionnement et donnons quelques notations. Les coûts étant asymétriques, le réseau est représenté par un graphe orienté $D = (V, A)$, avec V l'ensemble des sommets correspondant aux nœuds du réseau et A l'ensemble des arcs correspondant aux liens du réseau. La cardinalité de V est notée n. Les coûts associés aux arcs de A sont définis par le vecteur $c \in \mathbb{R}^A$. La flotte de véhicules F est supposée homogène. Les véhicules ont donc tous la même capacité, notée Q, et le même dépôt, noté v_0. L'ensemble des demandes est noté K et sa cardinalité p. On suppose, sans perte de généralité, que $K = \{1, 2, \ldots, p\}$. À chaque demande $k \in K$ est associé un couple (o^k, d^k), où o^k et d^k appartiennent à $V \setminus \{v_0\}$ et correspondent respectivement à l'origine et à la destination de la demande k. De plus, le volume de chaque demande $k \in K$, correspondant à la quantité devant être transportée de l'origine o^k à la destination d^k, est noté q^k. Aucune hypothèse n'est envisagée pour l'ensemble des demandes K. Un sommet peut donc être incident à zéro ou plusieurs demandes (les sommets différents du dépôt et incidents à aucune demande peuvent toujours intervenir pour les rechargements, comme montré dans l'exemple donné par la figure 2.1). Les rechargements peuvent être effectués en n'importe quel sommet du graphe, excepté le sommet v_0.

Dans ce chapitre, nous proposons une formulation du m-PRLAPF basée sur un graphe spatio-temporel. Une telle approche a été utilisée pour formuler d'autres problèmes dans [1, 45, 95]. L'utilisation d'un graphe spatio-temporel permet de représenter les solutions à l'aide d'un ensemble de chemins élémentaires. Nous donnons tout d'abord la construction du graphe auxiliaire en fonction du graphe D représentant le réseau. Nous montrons ensuite que l'utilisation de ce graphe permet de vérifier facilement si une solution du problème et réalisable. Nous donnons deux formulations pour le m-PRLAPF à l'aide de deux programmes linéaires mixtes basés sur ce graphe. Nous décrivons par la suite un algorithme de coupes et branchements pour résoudre les deux formulations puis discutons des résultats expérimentaux obtenus.

3.1 Programmes linéaires mixtes

Nous introduisons dans cette section un graphe auxiliaire à partir du graphe représentant le réseau, afin de pouvoir représenter toute solution par un ensemble de chemins élémentaires. Le trajet de chaque véhicule sera ainsi représenté dans le graphe auxiliaire par un chemin élémentaire et le trajet de chaque demande par un ensemble de

chemins élémentaires puisque chaque demande peut être transportée sur plusieurs chemins. Nous montrons que cette nouvelle représentation du problème permet de vérifier facilement si une solution donnée est réalisable. Nous donnons ensuite deux formulations linéaires mixtes basées sur ce nouveau graphe. La première exprime les chemins des demandes par un ensemble de variables associées aux arcs du graphe auxiliaire et aux demandes et considère des contraintes de conservation de flot. La deuxième remplace ces variables et ces contraintes par des contraintes métriques.

3.1.1 Graphe auxiliaire

L'introduction des rechargements et du fractionnement dans le transport des demandes modifie profondément la structure des solutions du problème de ramassage et livraison multi-véhicules asymétrique. Comme nous l'avons montré dans le chapitre 2, les trajets des véhicules correspondent à un ensemble de circuits non-élémentaires qui ne sont plus arc-disjoints. Le trajet de chaque demande est représenté par un ensemble de chemins élémentaires non arc-disjoints. De plus, à chaque chemin élémentaire est associée une valeur correspondant à la quantité de la demande transportée sur ce chemin. Il devient alors moins trivial de savoir si une solution donnée est réalisable pour le m-PRLAPF. Il faut notamment vérifier qu'il existe un ordre sur les arcs des circuits des véhicules conforme aux circuits des véhicules et aux chemins des demandes. Cette question de réalisabilité d'une solution est au cœur de la difficulté du m-PRLAPF. Dans ce chapitre, nous utilisons un graphe auxiliaire permettant de représenter les solutions à l'aide d'un ensemble de chemins élémentaires (mais non arc-disjoints). Une telle représentation nous permet de répondre facilement à la question de réalisabilité d'une solution comme nous le montrerons à la fin de cette partie. Nous commençons ici par décrire le graphe auxiliaire construit à partir du graphe D. Afin de distinguer les deux graphes considérés, nous appellerons le graphe D, représentant le réseau, *graphe initial*, en opposition au graphe auxiliaire.

Nous considérons un *graphe auxiliaire*, noté dans la suite par $\bar{D} = (\bar{V}, \bar{A})$, correspondant à un graphe spatio-temporel (des graphes similaires peuvent être trouvés dans [1] et [45]), dans lequel chaque sommet du graphe D, excepté le dépôt, est représenté à chaque unité de temps. On suppose ici que le temps est discret. Pour obtenir un graphe spatio-temporel fini, on considère une borne temporelle $T \in \mathbb{Z}_+$ à laquelle toutes les demandes doivent être acheminées et les véhicules être de retour à leur dépôt. Il est clair que la borne T peut être choisie aussi grande que nécessaire afin de garder l'ensemble initial des solutions. Afin de construire le graphe auxiliaire, on suppose que les arcs du graphe initial D sont munis de durées correspondant à des entiers strictement

positifs. La durée de chaque arc $a \in A$ est notée l_a et représente la durée nécessaire pour qu'un véhicule traverse l'arc a.

La détermination de la valeur de T la plus petite préservant l'ensemble des solutions réalisables du problème est un problème NP-difficile. En effet, si la flotte est composée d'un unique véhicule, cela revient à déterminer le trajet de coût minimum où les coûts correspondent aux durées des arcs. La borne T peut être déterminée de manière heuristique. Cependant, dans le cas général, il n'est pas évident de déterminer une "bonne" valeur de T. Cela dépend beaucoup des propriétés satisfaites par les durées des arcs ; par exemple si celles-ci sont symétriques, si elles vérifient les inégalités métriques ou si la durée de chaque arc est proportionnelle au coût de cet arc. La borne T peut par ailleurs être fixée arbitrairement afin d'éliminer les solutions ayant une durée de parcours des véhicules trop longue.

Le graphe $\bar{D}$ est construit de la manière suivante. Pour chaque sommet $u \in V \setminus \{v_0\}$, on associe $T-1$ sommets $u_1, u_2, \ldots, u_{T-1}$ dans $\bar{D}$. Le sommet u_t représente le sommet $u \in V \setminus \{v_0\}$ au temps $t \in \{1, \ldots, T-1\}$. Notons par V' cet ensemble de sommets. On ajoute également deux nouveaux sommets O^+ et O^- qui représentent respectivement le dépôt v_0 des véhicules à la date 0 et à la date T. On remarque qu'il est possible de séparer le dépôt du véhicule des autres sommets de D car les hypothèses retenues assurent que les véhicules ne passent par leur dépôt qu'au départ et à l'arrivée de leur trajet, c'est-à-dire, à la date 0 et à la date T. Pour chaque demande k de K, les sommets o_1^k et d_{T-1}^k représentent respectivement l'origine et la destination de la demande k.

L'ensemble d'arcs $\bar{A}$ est formé de quatre sous-ensembles d'arcs. Le premier, noté A_T, correspond à l'ensemble d'arcs $A_T = \{(u_t, u_{t+1}) \mid u \in V \setminus \{v_0\},\ t \in \{1, \ldots, T-2\}\}$ dans $\bar{D}$. Un arc de A_T correspond au fait qu'un véhicule ou une demande reste sur un sommet pour une unité de temps. Les arcs de A_T ont un coût nul puisque nous ne considérons aucun coût relatif au fait de rester à un même endroit.
On considère un deuxième ensemble d'arcs $\tilde{A} = \{(u_t, u'_{t+l_{(u,u')}}) \mid (u,u') \in A \setminus \delta(v_0),\ t \in \{1, \ldots, T-1-l_{(u,u')}\}\}$. Un arc $a = (u_t, u'_{t'}) \in \tilde{A}$ correspond au trajet d'un véhicule partant du sommet u au temps t et arrivant au sommet u' au temps t' avec $t' = t + l_{(u,u')}$. Chaque arc $a = (u_t, u'_{t'}) \in \tilde{A}$ a un coût égal à $c_{(u,u')}$.
Pour tout arc $(v_0, u) \in A$, on ajoute dans $\bar{D}$ l'arc (O^+, u_t), avec $t = l_{(v_0,u)}$, de coût égal à celui associé à (v_0, u). De la même manière, pour tout arc (u, v_0), on ajoute l'arc $(u_{t'}, O^-)$, avec $t' = T - l_{(u,v_0)}$, ayant le même coût que (u, v_0). Soient A^{O^+} l'ensemble des arcs sortant de O^+ et A^{O^-} l'ensemble des arcs entrant dans O^-. On a alors construit un graphe auxiliaire $\bar{D} = (\bar{V}, \bar{A})$ avec $\bar{V} = V' \cup \{O^+, O^-\}$ et $\bar{A} = A_T \cup \tilde{A} \cup A^{O^+} \cup A^{O^-}$.

On remarque qu'il est possible de restreindre le graphe auxiliaire $\bar{D}$. En effet, si un

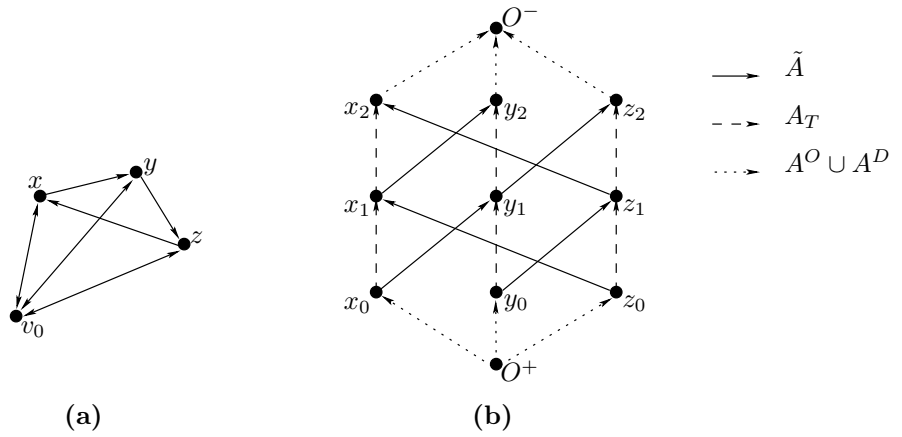

FIGURE 3.1 – Graphe initial et le graphe auxiliaire associé

sommet de D différent du dépôt, disons v, ne peut être atteint à partir du dépôt avant une certaine date t, on peut dans ce cas supprimer toutes ses copies dans $\bar{D}$ avant cette date, ainsi que tous les arcs incidents à ces sommets. De la même manière, si le dépôt ne peut être atteint en moins de t unités de temps à partir de v, les copies de v après la date $T - t$ et les arcs incidents peuvent être supprimées. Si le sommet v est l'origine (respectivement la destination) d'une demande, on considère le premier (respectivement dernier) sommet dans $\bar{D}$ associé à v comme origine (respectivement destination) pour cette demande.

Pour illustrer la construction du graphe auxiliaire, on considère le graphe initial $D = (V, A)$ donné dans la figure 3.1(a). On suppose que tous les arcs $a \in A$ ont une durée l_a égale à un et T est égal à quatre. Le graphe auxiliaire $\bar{D}$ correspondant au graphe D est donné dans la figure 3.1(b). Les différents ensembles d'arcs constituant l'ensemble $\bar{A}$ sont représentés par des types d'arcs différents.

Nous terminons cette section par la représentation des solutions sur le graphe auxiliaire et donnons les conditions pour qu'une solution soit réalisable. Il est clair que les circuits des véhicules dans le graphe D correspondent à des chemins élémentaires de O^+ à O^- dans $\bar{D}$. De plus, chaque chemin d'une demande $k \in K$ dans D peut être représenté par un chemin élémentaire de o^k à la date 1 à d^k à la date $T - 1$ dans $\bar{D}$.

Une telle représentation des solutions du m-PRLAPF permet de vérifier facilement qu'une solution du problème est réalisable. En effet, par construction, un chemin élémentaire associé au trajet d'un véhicule dans $\bar{D}$ correspond à un circuit passant par le dépôt v_0. De même, les chemins élémentaires des demandes correspondent à des chemins allant de o^k à d^k dans le graphe D pour toute demande $k \in K$. (Nous remarquons

que le chemin d'une demande dans le graphe D obtenu à partir d'un chemin élémentaire dans $\bar{D}$ n'est pas forcément élémentaire. Ceci n'a cependant aucune importance puisque nous pouvons en déduire un chemin élémentaire en supprimant les circuits.) De plus dans le graphe D, les chemins des demandes utilisent uniquement des arcs traversés par les véhicules si et seulement si dans $\bar{D}$, les chemins des demandes utilisent uniquement des arcs de A_T ou des arcs traversés par les véhicules. La condition imposant que la somme des quantités transportées sur les chemins élémentaires d'une demande soit égale au volume et celle forçant les véhicules à ne pas transporter plus de demande que leur capacité à tout instant est facilement vérifiable dans les deux graphes. Finalement, la construction du graphe auxiliaire nous assure que pour toute solution sur le graphe D obtenue à partir d'une solution sur $\bar{D}$, il existe un ordre sur les arcs traversés par les véhicules conforme aux chemins des demandes et aux circuits des véhicules. En effet, pour une solution donnée dans le graphe $\bar{D}$, nous définissons un ordre arbitraire des arcs de $\bar{A} \setminus A_T$ traversés par les véhicules tel que les arcs de A^{0^+} (respectivement A^{O^-}) sont avant (respectivement après) tous les autres arcs et les arcs commençant à la date t (*i.e.*, les arcs dont le sommet initial correspond à un sommet de $V \setminus \{v_0\}$ à la date t) sont avant tous les arcs commençant à la date $t' > t$, pour toute date $t \in \{1, 2, \ldots, T-2\}$. Il est clair que l'ordre ainsi obtenu est conforme à tout chemin élémentaire dans le graphe $\bar{D}$ puisque ce dernier est acyclique. Il est évident que cet ordre est aussi conforme aux chemins des demandes et aux circuits des véhicules lorsque l'on transpose la solution dans D.

Nous pouvons remarquer que les chemins des véhicules correspondent, dans le graphe auxiliaire, à un ensemble de chemins élémentaires partant de O^+ et arrivant en O^-. Comme la capacité des véhicules est la même, il n'est pas nécessaire de connaître à quel véhicule correspond chaque chemin. Il suffit que l'on ait un nombre de chemins inférieur ou égal à la cardinalité de la flotte de véhicules. Nous pouvons donc représenter les trajets des véhicules par un ensemble d'arcs représentant un ensemble de chemins élémentaires.

3.1.2 Formulation multiflots

Nous donnons maintenant une formulation pour le m-PRLAPF à l'aide d'un programme linéaire mixte utilisant les contraintes de conservation de flot pour représenter les chemins des demandes. Comme mentionné dans la partie précédente, il est possible de ne pas différencier les chemins des véhicules. Nous pouvons considérer un ensemble de chemins de cardinalité au plus $|F|$ représentant l'ensemble des trajets des véhicules. Dans cette première formulation, nous considérons aussi un multiflot qui représente

les chemins des demandes dans le réseau. Nous considérons alors les deux types de variables suivantes :

- $y \in \mathbb{Z}_+^{|\bar{A}|}$ où y_a représente le nombre de véhicules passant sur l'arc $a \in \bar{A}$,
- $x \in \mathbb{R}_+^{p|\tilde{A} \cup A_T|}$ où x_a^k représente la quantité de demande $k \in K$ transportée sur l'arc $a \in \tilde{A} \cup A_T$.

On remarque qu'une variable y_a est associée à chaque arc $a \in A_T$ car un véhicule peut rester sur un nœud du réseau pendant plusieurs unités de temps, ce qui correspond, dans le graphe auxiliaire $\bar{D}$, à emprunter un chemin composé d'arcs de A_T. Pour un sommet u_t avec $u \in V \setminus \{v_0\}$ et $t \in \{1, \ldots, T-1\}$, et une demande $k \in K$, le nombre $b_{u_t}^k$ défini par

$$b_{u_t}^k = \begin{cases} q^k & \text{si} \quad u = o^k \text{ et } t = 1, \\ -q^k & \text{si} \quad u = d^k \text{ et } t = T-1, \\ 0 & \text{sinon,} \end{cases}$$

représente le volume de la demande k disponible/requis au sommet u_t. On remarque que pour une demande $k \in K$, il y a exactement deux sommets de V' pour lesquels $b_{u_t}^k$ est non nul. Ces deux sommets sont associés au sommet origine de la demande k au temps 1 et au sommet destination au temps $T-1$. (Nous avons déjà montré que le trajet d'une demande était représenté par un chemin élémentaire allant de l'origine de la demande à la date 1 à sa destination à la date $T-1$.) L'exactitude de la définition de $b_{u_t}^k$ est impliquée par l'équivalence entre le fait qu'une demande (ou partie) reste sur un même sommet du réseau et utilise un chemin dans $\bar{D}$ composé d'arcs de A_T.

Le problème de ramassage et livraison multi-véhicules asymétrique préemptif avec fractionnement peut être formulé comme un programme linéaire mixte utilisant une approche arc-sommet [1] comme suit.

$$\min \sum_{a \in \bar{A}} c_a y_a$$

s.t.

$$\sum_{a \in A^{O+}} y_a \leq |F|, \tag{3.1}$$

$$\sum_{a \in \delta^{\text{out}}(v)} y_a - \sum_{a \in \delta^{\text{in}}(v)} y_a = 0 \qquad \forall\, v \in V', \tag{3.2}$$

$$\sum_{a \in \delta^{\text{out}}(v) \setminus A^{O-}} x_a^k - \sum_{a \in \delta^{\text{in}}(v) \setminus A^{O+}} x_a^k = b_v^k \qquad \forall\, k \in K,\ \forall\, v \in V', \tag{3.3}$$

$$\sum_{k \in K} x_a^k - Q y_a \leq 0 \qquad \forall\, a \in \tilde{A}, \tag{3.4}$$

$$x_a^k \geq 0 \qquad \forall\, a \in \tilde{A} \cup A_T,\ \forall\, k \in K, \tag{3.5}$$

$$y_a \geq 0 \qquad \forall\, a \in \bar{A}, \tag{3.6}$$

$$y_a \text{ entier} \qquad \forall\, a \in \bar{A}, \tag{3.7}$$

La fonction objectif signifie que le coût total des chemins des véhicules doit être minimum. (Nous remarquons que la fonction peut être facilement étendue en considérant des coûts proportionnels à la quantité de demande transportée sur chaque arc.) La contrainte (3.1) limite le nombre de véhicules disponibles à $|F|$. Les contraintes (3.2) (resp. (3.3)) sont les contraintes de conservation de flot associées aux véhicules (resp. demandes) et impliquent que les trajets des véhicules et des demandes soient des chemins (l'élémentarité de ces chemins dans le graphe auxiliaire provient du fait que ce dernier est acyclique). Les contraintes (3.4) imposent que la quantité de demande transportée sur un arc de $\tilde{A}$ ne dépasse pas la capacité totale des véhicules traversant cet arc. Les autres contraintes sont les contraintes triviales et les contraintes d'intégrité. Nous remarquons que les contraintes (3.6) ne sont pas nécessaires puisque les contraintes (3.4) et (3.5) impliquent déjà que y est positif ou nul. Ce modèle contient $|\bar{A}| + p \times (|\tilde{A}| + |A_T|)$ variables et $|V'|(p+1) + |\bar{A}| + p \times (|\tilde{A}| + |A_T|) + |\tilde{A}| + 1$ contraintes.

Cette formulation peut facilement être étendue pour prendre en compte les contraintes de fenêtres de temps. En effet, si une demande $k \in K$ doit être chargée à son origine o^k dans l'intervalle $[e_i, b_i]$ où e_i et b_i sont des valeurs entières supérieures à zéro, il suffit d'ajouter les contraintes $x^k_{(o^k_t, o^k_{t+1})} = 1$ (respectivement $x^k_{(o^k_t, o^k_{t+1})} = 0$) pour toute date $t \in \{1, 2, \ldots, e_i - 1\}$ (respectivement $t \in \{b_i, b_i + 1, \ldots, T - 1\}$). Nous avons évidemment les mêmes types de contraintes si une fenêtre de temps est associée à la destination d^k de la demande k.

3.1.3 Formulation métrique

Dans la formulation précédente, nous déterminons les trajets des demandes alors que la valeur de la fonction objectif dépend uniquement des trajets des véhicules et qu'aucune contrainte additionnelle sur le trajet des demandes n'est considérée. Il serait donc suffisant de déterminer uniquement les trajets des véhicules permettant de transporter les demandes de leur origine à leur destination. Ceci peut être fait en ne considérant pas d'autres variables que les variables y_a définies pour tout $a \in \bar{A}$ et en remplaçant les contraintes (3.3) et (3.4) par les contraintes métriques décrites ci-dessous.

Iri [51], et Onaga et Kakhuso [75] ont indépendamment montré que les contraintes métriques peuvent être utilisées pour vérifier s'il existe un multiflot réalisable quand les demandes et les capacités sont données. Leur résultat, connu sous le nom du théorème japonais, peut être brièvement décrit comme suit. Soit $\hat{D} = (\hat{V}, \hat{A})$ un graphe orienté complet et $w \in \mathbb{R}_+^{|\hat{A}|}$ (resp. $r \in \mathbb{R}_+^{|\hat{A}|}$) le vecteur capacité (resp. demande) indexé sur les arcs de $\hat{A}$. Le vecteur capacité w permet le transport des demandes de r si et seulement si toutes les contraintes métriques

$$(w - r)^T \pi \geq 0 \quad \forall\, \pi \in Met_n \tag{3.8}$$

sont satisfaites où $Met_n = \{\pi \in \mathbb{R}_+^{\overline{A}} \mid \pi_{ik} + \pi_{kj} - \pi_{ij} \geq 0 \ \forall i \neq j \neq k \neq i\}$ est le *cône métrique*. Clairement, le nombre de contraintes métriques est infini. Cependant, celles associées aux rayons extrêmes du cône métrique sont suffisantes pour s'assurer que le vecteur capacité w permet le transport des demandes de r. Ces contraintes sont cependant en nombre exponentiel.

Pour notre problème, le cône métrique est celui induit par le graphe complet sur V', et les vecteurs capacité et demande sont donnés par

$$w_a = \begin{cases} +\infty & \text{si } a \in A_T, \\ y_a & \text{si } a \in \tilde{A}, \\ 0 & \text{sinon,} \end{cases}$$

et

$$r_a = \begin{cases} \frac{q^k}{Q} & \text{si } a = (u_0, w_{T-1}) \text{ avec } u = o^k \text{ et } w = d^k \text{ pour un } k \in K, \\ 0 & \text{sinon,} \end{cases}$$

pour tout arc du graphe complet sur V' (nous remarquons que nous aurions pu considérer les capacités et demandes totales pour les véhicules et les demandes). En utilisant les contraintes métriques (3.8) à la place des contraintes (3.3) et (3.4), nous introduisons maintenant la formulation linéaire entière suivante pour le problème de ramassage et livraison multi-véhicules asymétrique préemptif avec fractionnement :

$$\min\{\sum_{a\in\bar{A}} c_a y_a \mid y \text{ satisfait } (3.1), (3.2), (3.8), (3.6), (3.7)\} \tag{3.9}$$

Cette formulation contient moins de variables que la première (*i.e.*, $|\bar{A}|$ contre $|\bar{A}| + p|\tilde{A} \cup A_T|$), mais possède un nombre exponentiel de contraintes alors que le premier modèle en contient un nombre polynomial. Cependant, le nombre exponentiel de contraintes métriques peut être traité en temps polynomial comme indiqué dans la section 3.3.

Par ailleurs, une fois que nous avons une solution optimale de (3.9), déterminer les trajets des demandes (*i.e.*, les variables x_a^k pour tout $a \in \tilde{A} \cup A_T$ et pour tout $k \in K$ dans la première formulation) peut être résolu en temps polynomial. Ce problème n'est en fait rien d'autre que la résolution d'un problème de multiflots (non-entier) qui est un problème polynomial [1].

Les deux formulations peuvent facilement être étendues à une variante du problème de ramassage et livraison préemptif multi-véhicules avec fractionnement pour prendre en compte des contraintes de capacité sur le volume des demandes stockées à n'importe quel endroit du réseau à tout instant. En effet, si nous notons par e_u la capacité de stockage des demandes au sommet $u \in V$, il suffit alors d'ajouter les contraintes $\sum_{k\in K} x_a^k \leq e_u$ pour tout $a = (u_t, u_{t+1})$ avec $t \in \{1, \ldots, T-2\}$, dans la formulation multiflots. Pour considérer de telles contraintes dans la formulation métrique, nous devons modifier le vecteur capacité w en remplaçant la capacité infinie associée aux arcs de A_T par la valeur $w_a = \frac{e_u}{Q}$ pour tout arc $a = (u_t, u_{t+1})$ avec $t \in \{1, \ldots, T-2\}$.

3.2 Contraintes valides

Nous présentons dans cette section trois familles de contraintes que nous utilisons dans notre algorithme pour renforcer les relaxations linéaires. La première peut être utilisée pour les deux formulations alors que la deuxième famille de contraintes est basée sur les variables de flot associées aux demandes (*i.e.*, les variables x) et sera donc uniquement utilisée pour la formulation multiflots. La dernière famille de contraintes

est une formulation renforcée des contraintes métriques (3.8) et ne sera donc utilisée que dans la seconde formulation. La séparation de ces contraintes sera décrite dans la section suivante.

3.2.1 Contraintes de coupes

Nous introduisons des contraintes connues dans la littérature sous le nom de contraintes de coupes que nous utiliserons dans nos algorithmes de coupes et branchements pour renforcer les deux relaxations linéaires du problème de ramassage et livraison multi-véhicules asymétrique préemptif avec fractionnement. Ces contraintes ont déjà été utilisées dans de nombreux autres problèmes d'optimisation [11, 18]. Elles sont basées sur la notion de coupes dans le graphe auxiliaire $\bar{D}$. Pour un sous-ensemble de sommets $W \subset V'$, $W \neq \emptyset$, nous notons par $q[W, V' \setminus W]$ le volume total des demandes de K ayant leur sommet d'origine dans W et leur destination dans $V' \setminus W$.

Nous ne considérons que les coupes n'intersectant pas l'ensemble d'arcs A_T car il n'existe pas de contraintes de capacité liées à ces arcs. (Dans la première formulation, il n'existe pas de contraintes (3.4) associées à ces arcs et dans la seconde, nous considérons une capacité infinie pour chacun de ces arcs.) Cela s'explique par le fait qu'une demande (ou partie) peut rester sur un noeud du réseau.

Proposition 3.1 *Soit $W \subset \bar{V}$, $W \neq \emptyset$ un sous-ensemble de sommets tel que la coupe sortante induite n'intersecte aucun arc de A_T, i.e. $\delta^{\text{out}}(W) \cap A_T = \emptyset$. La contrainte de coupe associée à W définie par*

$$y(\delta^{\text{out}}(W)) \geq \left\lceil \frac{q[W, V' \setminus W]}{Q} \right\rceil \tag{3.10}$$

est valide pour le m-PRLAPF.

Preuve : La contrainte $Qy(\delta^{\text{out}}(W)) \geq q[W, V' \setminus W]$ est clairement valide pour le m-PRLAPF. En effet, elle exprime le fait que la capacité totale des véhicules sur les arcs de $\delta^{\text{out}}(W)$ soit supérieure ou égale au volume total des demandes allant de W à $V' \setminus W$. Comme y est entier, en divisant les deux membres de la contrainte par Q et en prenant la valeur entière supérieure du membre de droite, on obtient la contrainte (3.10).□

Il est clair que l'on peut restreindre l'ensemble des sommets W à ceux contenant le sommet O^- mais pas le sommet O^+. En effet, ces deux sommets n'ont respectivement pas d'arcs sortants ni d'arcs entrants et aucun d'eux n'est incident à une demande.

3.2.2 Contraintes de capacité résiduelle

Les contraintes de capacité résiduelles ont été introduites par Magnanti et al. [64] pour des problèmes de conception de réseaux avec capacités entières sur les arcs et sont un cas particulier des contraintes mixte-entières arrondies [71]. Elles relient la capacité (entière) d'un arc par rapport à la quantité de flot (réel) traversant cet arc.

Avant d'exprimer les contraintes de capacité résiduelle, on donne quelques notations. Étant donné un sous-ensemble K' de K, le volume total des demandes $k \in K'$ est égal à $q(K') = \sum_{k \in K'} q^k$. Soit $\gamma_{K'} = \lceil \frac{q(K')}{Q} \rceil$ et $r_{K'} = (q(K') \; mod \; Q)$. Par convention, nous posons que $r_{K'}$ est égal à Q si $q(K')$ est un multiple de Q (*i.e.*, $r_{K'} \in]0, Q]$.) Nous donnons pour le problème les contraintes de capacité résiduelle sans donner la preuve de la validité de ces contraintes puisque la preuve est similaire à celle donnée par Magnanti et al. [64] lorsqu'ils ont introduit ces contraintes.

Proposition 3.2 *Soit $K' \subseteq K$ et $a \in \tilde{A}$. La contrainte de capacité résiduelle*

$$\sum_{k \in K'} x_a^k - r_{K'} \; y_a \leq (\gamma_{K'} - 1)(Q - r_{K'}) \tag{3.11}$$

est valide pour le m-PRLAPF. □

Dans ce qui suit, nous donnons un exemple d'une contrainte de capacité résiduelle violée. Supposons que l'on ait deux demandes ayant comme volume $q^1 = 6$ et $q^2 = 12$. Supposons également que la capacité Q du véhicule soit égale à 10. Soit $(\overline{x}, \overline{y})$ une solution fractionnaire satisfaisant les contraintes (3.1)-(3.6) telle qu'il existe un arc $a' \in \tilde{A}$ avec $x_{a'}^k = q^k$ pour $k = 1, 2$ et $y_{a'} = 1.8$. Il existe une contrainte de capacité résiduelle associée à l'arc a' qui est violée par $(\overline{x}, \overline{y})$. En effet, si nous considérons l'ensemble de demandes $K' = K$, nous avons $\gamma_{K'} = 2$ et $r_{K'} = 8$, et la contrainte implique que nous devons avoir $8y_{a'} \geq (6 + 12) - (2 - 1)(10 - 8) = 16$, c'est-à-dire, $y_{a'} \geq 2$.

3.2.3 Contraintes étendues de capacité résiduelle

Les contraintes de capacité résiduelle (3.11) sont associées à un arc du graphe. On montre maintenant que ces contraintes peuvent être étendues à un ensemble d'arcs lorsque le graphe est un graphe spatio-temporel. Pour cela, nous présentons quelques notations. Pour une valeur $\phi \in \{1, \ldots, T-2\}$ donnée, nous définissons $V^\phi = \{u_t \in$

$V' \mid u \in V \setminus \{v_0\}, t = \phi\}$ et $A^\phi = \delta^{\text{out}}(V^\phi) \cap \tilde{A}$. V^ϕ correspond aux sommets de $V \setminus \{v_0\}$ au temps ϕ et A^ϕ correspond aux arcs de $\tilde{A}$ dont l'extrémité initiale est un sommet de V^ϕ.

Proposition 3.3 *Soient $K' \subseteq K$, $\phi \in \{1, \ldots, T-2\}$ et $X \subseteq A^\phi$. La contrainte étendue de capacité résiduelle associée à K', ϕ et X*

$$\sum_{k \in K'} \sum_{a \in X} x_a^k - r_{K'} \sum_{a \in X} y_a \leq (\gamma_{K'} - 1)(B - r_{K'}) \tag{3.12}$$

est valide pour le m-PRLAPF.

Preuve. Nous remarquons tout d'abord que le graphe auxiliaire est acyclique, *i.e.*, il ne contient aucun arc (u, w) avec $u \in V^{\phi_1}$, $w \in V^{\phi_2}$ et $\phi_2 \leq \phi_1$. Comme $x_a^k \leq q^k$ pour tout $a \in \tilde{A} \cup A_T$, nous avons $\sum_{a \in \delta^{\text{out}}(V^\phi)} x_a^k \leq q^k$ pour tout $\phi \in \{1, \ldots, T-2\}$ et pour tout $k \in K$. Soient $\phi \in \{1, \ldots, T-2\}$, $K' \subseteq K$ et $X \subseteq A^\phi$. Nous pouvons maintenant montrer que la contrainte (3.11) est dominée par la somme des contraintes $\sum_{a \in X} x_a^k \leq q^k$ pour tout $k \in K'$ (resp. les contraintes (3.4) pour tout arc dans X) si $\sum_{a \in X} y_a \geq \gamma_{K'}$ (resp. $\sum_{a \in X} y_a \leq \gamma_{K'} - 1$.) Les contraintes étendues de capacité résiduelle (3.12) sont donc valides pour le m-PRLAPF. □

Il est clair que les contraintes étendues de capacité résiduelle ne peuvent être utilisées pour la formulation métrique puisqu'elles sont définies par rapport aux variables x qui n'apparaissent pas dans cette formulation. Elles sont par contre utilisées pour renforcer la relaxation linéaire de la formulation multiflots.

3.2.4 Contraintes métriques arrondies

Les contraintes métriques (3.8) peuvent être renforcées en tenant compte du fait que les variables y sont entières, si la métrique π satisfait quelques conditions. En effet, considérons une contrainte métrique associée à une métrique π ayant une valeur π_a entière pour tout arc $a \in \tilde{A}$. Comme on a $w_a = y_a$ pour tout arc $a \in \tilde{A}$, alors $\sum_{a \in \tilde{A}} w_a \pi_a$ est entier. Comme w_a a pour valeur 0 ou $+\infty$ pour tout arc a n'appartenant pas à $\tilde{A}$, il est possible d'arrondir supérieurement, comme pour les contraintes de coupes, la partie de la contrainte relative aux demandes. On obtient la contrainte métrique arrondie

$$w^T \pi \geq \lceil r^T \pi \rceil \tag{3.13}$$

qui est valide pour le m-PRLAPF. Des contraintes similaires ont été utilisées pour résoudre le Problème d'Installation de Capacité de Coût Minimum (PICCM) [18].

On peut remarquer que les contraintes de coupes (3.10) correspondent à un cas particulier des contraintes métriques arrondies. En effet, la contrainte de coupe associée à un ensemble W (avec $O^- \in W$ et $O^+ \notin W$) peut être obtenue par la contrainte métrique arrondie associée à la métrique π définie par

$$\pi_a = \begin{cases} 1 & \text{si } a \in \delta^{\text{out}}(W), \\ 0 & \text{sinon.} \end{cases}$$

3.3 Séparation

Dans cette section, nous considérons les problèmes de séparation associées aux contraintes métriques (3.8) et aux contraintes métriques arrondies (3.13), aux contraintes de coupes (3.10) et aux contraintes étendues de capacité résiduelle (3.12). Les contraintes de capacité résiduelle (3.11) sont séparées en même temps que les contraintes étendues de capacité résiduelle. Les solutions que nous considérons dans les problèmes de séparation sont les vecteurs $\overline{y} \in \mathbb{R}^{|\bar{A}|}$ pour la formulation métrique et les vecteurs $(\overline{x}, \overline{y}) \in \mathbb{R}^{p|\tilde{A} \cup A_T|} \times \mathbb{R}^{|\bar{A}|}$ pour la formulation multiflots que nous obtenons après avoir résolu la relaxation linéaire courante de l'algorithme de coupes et branchements.

3.3.1 Séparation des contraintes métriques et des contraintes métriques arrondies

Le problème de séparation des contraintes métriques peut être résolu en temps polynomial contrairement à celui des contraintes métriques arrondies [18]. Pour cela, la séparation des contraintes métriques et des contraintes métriques arrondies est faite dans une même procédure de séparation. Dans cette dernière, on résoud de manière exacte le problème de séparation associé aux contraintes métriques. S'il existe une contrainte métrique violée par la solution $\overline{y}$, alors une heuristique essaie de renforcer cette inégalité métrique violée en une inégalité métrique arrondie violée. Si cette tentative réussit, alors la contrainte métrique arrondie est ajoutée au programme linéaire courant. Autrement, l'inégalité métrique est ajoutée. Si aucune contrainte métrique n'est violée par la solution $\overline{y}$, la recherche de contrainte métrique arrondie violée n'est pas effectuée. Dans ce qui suit, on présente la séparation exacte des contraintes métriques

puis on explique ensuite la recherche d'une contrainte métrique arrondie à partir d'une contrainte métrique violée.

Le problème de séparation des contraintes métriques (3.8) peut être résolu en temps polynomial. En fait, ce dernier se ramène à résoudre le programme linéaire suivant :

$$\begin{aligned} &\min(w-r)^T\pi \\ &\text{s.t.} \\ &\pi \in Met_n, \end{aligned}$$

où Met_n, w et r sont définis comme dans la sous-section 3.1.3. Soit π^* une solution optimale de ce programme linéaire. Si la valeur de la fonction objectif associée à π^* est strictement négative, la contrainte métrique $(w-r)^T\pi^* \geq 0$ est alors violée par $\overline{y}$. Dans le cas contraire, on peut certifier que $\overline{y}$ satisfait toutes les contraintes métriques.

Supposons maintenant que la solution $\overline{y}$ viole une contrainte (3.8) associée à une métrique π. Si les coefficients π_a associés aux arcs de $\tilde{A}$ sont entiers, alors la contrainte métrique trouvée peut directement être étendue en une contrainte métrique arrondie en arrondissant supérieurement la valeur $r^T\pi$. Dans le cas contraire, on cherche parmi les coefficients métriques π_a, $a \in \tilde{A}$, celui ayant la plus petite valeur strictement positive et l'on vérifie si les valeurs $\frac{\pi_a}{\pi_0}$ pour tout $a \in \tilde{A}$ sont entières. Si tel est le cas, la contrainte métrique arrondie $w^T\pi' \geq \lceil r^T\pi' \rceil$, avec $\pi' = \frac{\pi}{\pi_0}$ est violée par $\overline{y}$. Cette heuristique, développée par Bienstock et al. [18] (la seule modification consiste à limiter la recherche des coefficients entiers à ceux associés aux arcs de $\tilde{A}$ au lieu de tout l'ensemble d'arcs), peut être appliquée en temps linéaire.

3.3.2 Séparation des contraintes de coupes

Le problème de séparation associé aux contraintes de coupes (3.10) est NP-difficile dans le cas général [18]. Ces contraintes sont donc séparées de manière heuristique. Dans ce qui suit, nous décrivons trois heuristiques de séparation qui sont basées sur des travaux précédents [18, 41]. Nous adaptons ces heuristiques au graphe spatio-temporel. Dans ces descriptions, W représentera toujours un sous-ensemble de sommets de $\bar{V}$ tel que $\delta^{\text{out}}(W) \cap A_T = \emptyset$.

La première heuristique est l'*heuristique n-cut*, développée par Bienstock et al. [18] pour le PICCM. Cette heuristique, décrit par l'algorithme 1, fonctionne de la façon suivante. Pour chaque demande $k \in K$, on vérifie s'il existe un chemin de o_1^k à d_{T-1}^k (ces deux sommets correspondent respectivement à l'origine o^k de la demande k au

Algorithme 1: Heuristique n-Cut

Entrée : - vecteur $\overline{y}$
Sortie : - Ensemble S de contraintes de coupes violées par $\overline{y}$
Début
  $S = \emptyset$;
  Soit $\hat{D}$ le graphe induit par l'ensemble d'arcs $A^{O^-} \cup A_T \cup \{a \in \tilde{A} \mid \overline{y_a} > 0\}$;
  pour chaque *demande* $k \in K$ **faire**
    Parcours en profondeur à partir de o_1^k dans $\hat{D}$;
    Soit W l'ensemble des sommets visités lors du parcours;
    si $d_{T-1}^k \notin W$ **alors**
      Ajouter dans S la contrainte de coupe associée à W;
  si $S = \emptyset$ **alors**
    Choisir W aléatoirement tel que $\delta^{\text{out}}(W) \cap A_T = \emptyset$;
    si *la contrainte de coupe associée à* W *est violée par* $\overline{y}$ **alors**
      Ajouter dans S la contrainte de coupe associée à W ;
  retourner S;
Fin

temps 1 et à sa destination d^k au temps $T-1$) dans le graphe auxiliaire $\bar{D}$ où seuls les ensembles d'arcs $A^{O^-} \cup A_T \cup \{a \in \tilde{A} \mid \overline{y_a} > 0\}$ sont pris en compte. Cette vérification est faite à l'aide d'un parcours en profondeur à partir du sommet o_1^k. Si le parcours permet d'atteindre le sommet d_{T-1}^k, alors il existe un chemin entre l'origine et la destination de la demande k. Autrement, il est clair que la contrainte associée à l'ensemble W des sommets atteints à partir de o_1^k durant le parcours est violée. En effet, il est clair que l'on a $\overline{y}(\delta^{\text{out}}(W)) = 0$ et la demande k a son origine dans W et sa destination dans $V' \setminus W$, ce qui implique que $\left\lceil \frac{q[W,V'\setminus W]}{Q} \right\rceil > 0$. De plus, il est clair que $\delta^{\text{out}}(W) \cap A_T = \emptyset$ est garanti car tous les arcs de A_T sont considérés.
Si un chemin de o_1^k à d_{T-1}^k est trouvé pour toutes les demandes k de K, on choisit alors aléatoirement un ensemble W et l'on teste si la contrainte de coupe associée à W est violée ou non. Comme la complexité d'un algorithme de recherche est en $\mathcal{O}(|\tilde{A} \cup A_T|)$, et que p parcours sont appliqués, la complexité de cette heuristique est en $\mathcal{O}(p|\tilde{A} \cup A_T|)$. Bien que cette heuristique puisse être appliquée pour renforcer les deux formulations, elle n'a d'intérêt que pour la formulation métrique. En effet, dans la formulation multiflots, les contraintes de conservation de flot associées aux demandes (3.2) et les contraintes de capacité (3.4) assurent déjà qu'un chemin existe de l'origine (au temps 1) à la destination (au temps $T-1$) pour chaque demande dans le graphe considéré.

Algorithme 2: Heuristique n-Partition

Entrée : - vecteur $\overline{y}$
Sortie : - Ensemble S de contraintes de coupes violées par $\overline{y}$
Début
  $S = \emptyset$;
  Choisir aléatoirement une demande $k \in K$;
  $V_1 = \{o_t^k \mid t = 1, \ldots, T-1\}$;
  $V_2 = \{d_t^k \mid t = 1, \ldots, T-1\}$;
  $U = V \setminus \{v_0, o^k, d^k\}$;
  tant que $U \neq \emptyset$ **faire**
    Choisir aléatoirement un sommet $u \in U$;
    $U = U \setminus \{u\}$;
    Ajouter les sommets $\{u_t \mid t = 1, \ldots, T-1\}$ à V_1 et V_2 tel que $\delta^{\text{out}}(V_1) \cap A_T = \emptyset$ et $\overline{y}(\delta^{\text{out}}(V_1) \cap \delta^{\text{in}}(V_2)) - \lceil \frac{q[V_1,V_2]}{Q} \rceil$ est minimum;
  $V_1 = V_1 \cup \{O^-\}$;
  si *la contrainte de coupe associée à* V_1 *est violée par* $\overline{y}$ **alors**
    Ajouter dans S la contrainte de coupe associée à V_1;
  retourner S;
Fin

La seconde heuristique est basée sur l'*heuristique de n-partition* imaginée par Bienstock et al. [18]. On considère uniquement le cas n=2 étant donné que l'on recherche des contraintes de coupes (3.10). L'algorithme 2 donne l'algorithme de cette heuristique. On commence avec une demande $k \in K$ choisie aléatoirement et deux sous-ensembles de sommets de V', $V_1 = \{o_t^k \mid t = 1, \ldots, T-1\}$ et $V_2 = \{d_t^k \mid t = 1, \ldots, T-1\}$. On affecte ensuite itérativement les sommets de $\bar{V} \setminus (V_1 \cup V_2)$ à V_1 ou V_2 de manière à considérer, à la fin de l'heuristique, la contrainte de coupe induite par le sous-ensemble V_1. L'affectation des sommets restants aux sous-ensembles V_1 et V_2 se fait de la manière suivante. À chaque itération, on choisit aléatoirement un sommet $u \in V \setminus \{v_0, o^k, d^k\}$ qui n'a pas encore été considéré. On considère alors uniquement l'ensemble de sommets $V_1 \cup V_2 \cup \{u_t \mid t = 1, \ldots, T-1\}$ et l'on affecte les sommets $\{u_t \mid t = 1, \ldots, T-1\}$ à V_1 ou V_2, l'affectation des sommets de ces deux ensembles restant inchangée. L'affectation des sommets $\{u_t \mid t = 1, \ldots, T-1\}$ est appliquée de façon à obtenir V_1 et V_2 tel que $\delta^{\text{out}}(V_1) \cap A_T = \emptyset$ et $\overline{y}(\delta^{\text{out}}(V_1) \cap \delta^{\text{in}}(V_2)) - \lceil \frac{q[V_1,V_2]}{Q} \rceil$ est minimum. Ceci est fait par énumération de toutes les affectations possibles des sommets $\{v_t \mid t = 1, \ldots, T-1\}$ puisqu'il en existe seulement $T+1$ réalisables par rapport à la condition $\delta^{\text{out}}(V_1) \cap A_T = \emptyset$. Une fois que tous les sommets de $V \setminus \{v_0, o^k, d^k\}$ ont été considérés, on teste si la contrainte de coupe associée au sous-ensemble $V_1 \cup \{O^-\}$ est violée.

On discute maintenant de la complexité de l'heuristique. Il est clair cette complexité dépend de la complexité de l'opération d'ajout des sommets $\{u_t : t = 1, 2, \ldots, T-1\}$ aux sous-ensembles V_1 et V_2. Notons (V_1^i, V_2^{T-i}), $i = 0, 1, \ldots, T$, les couples tels que V_1^i correspond à V_1 augmenté des i sommets de $\{u_t : t = 1, 2, \ldots, T-1\}$ de plus grand indice et V_2^{T-i} est constitué de l'ensemble V_2 et des sommets de $\{u_t : t = 1, 2, \ldots, T-1\}$ qui n'appartiennent pas à V_1^i. Remarquons que la valeur $\overline{y}(\delta^{\text{out}}(V_1) \cap \delta^{\text{in}}(V_2)) - \lceil \frac{q[V_1,V_2]}{Q} \rceil$, notée par la suite $\rho(V_1, V_2)$, est connue au début de chaque itération lorsqu'aucun sommet u_t, $t = 1, 2, \ldots, T-1$ n'a encore été ajouté. Il est clair que la valeur $\rho(V_1^0, V_2^T)$ se calcule à partir de $\rho(V_1, V_2)$ en ajoutant la valeur $\overline{y}$ associée aux arcs de $\tilde{A}$ sortant de V_1 et entrant dans $\{u_t : t = 1, 2, \ldots, T-1\}$ et en retranchant la valeur des volumes des demandes $k \in K$ telles que $o^k \in V_1$ et $d^k = u$. (Le calcul de $\rho(V_1^T, V_2^0)$ se calcule de manière similaire.) Par ailleurs, pour tout $i = 1, 2, \ldots, T-1$, on a

$$\rho(V_1^i, V_2^{T-i}) = \rho(V_1^{i-1}, V_2^{T-i+1}) + \overline{y}(\delta^{\text{out}}_{\tilde{A}}(u_i) \cap \delta^{\text{in}}(V_2)) - \overline{y}(\delta^{\text{in}}_{\tilde{A}}(u_i) \cap \delta^{\text{out}}(V_1)).$$

On déduit que la complexité pour déterminer l'indice $i \in 0, 1, \ldots, T$ tel que $\rho(V_1^i, V_2^{T-i})$ est maximum se fait en $O(\max\{m_u, p_u\})$ où m_u et p_u correspondent respectivement aux nombres d'arcs de $\tilde{A}$ et de demandes incidentes à l'ensemble $\{u_t : t = 1, 2, \ldots, T-1\}$. Comme cette ajout est fait pour tout sommet $u \in V \setminus \{v_0, o^k, d^k\}$, il est clair que l'heuristique est en $O(\max\{\tilde{A}, p\})$.

La dernière heuristique est une extension de l'heuristique Max-Ratio-Cut développée par Gabrel et al. [41] pour le problème d'installation de capacité de coût minimum avec une fonction coût en escalier. Cette heuristique, donnée par l'algorithme 3 associe à chaque sous-ensemble $W \subseteq \bar{V}$ tel que $\delta^{\text{out}}(W) \cap A_T = \emptyset$, un ratio $\rho(W)$ défini par

$$\rho(W) = \frac{\left\lceil \frac{q[W, V' \setminus W]}{Q} \right\rceil}{\overline{y}(\delta^{\text{out}}(W))}.$$

L'algorithme part d'un sous-ensemble W choisi aléatoirement (W vérifie $o_1^k, O^- \in W$, $d_{T-1}^k, O^+ \in \bar{V} \setminus W$ et $\delta^{\text{out}}(W) \cap A_T = \emptyset$, avec $k \in K$ la demande passée en paramètre) et modifie itérativement ce sous-ensemble tant que cette modification augmente le ratio $\rho(W)$. Lorsqu'aucune modification n'augmente le ratio, l'heuristique teste si la contrainte de coupe associée à W est violée par $\overline{y}$.
La modification à chaque itération du sous-ensemble W, donnée par l'algorithme 4, s'effectue comme suit. Pour chaque sommet $v \in V \setminus \{v_0, o^k, d^k\}$, on considère les $T+1$ sous-ensembles, disons $W_v^0, W_v^1, \ldots, W_v^T$, obtenus à partir de W en échangeant entre W et $\bar{V} \setminus W$ les sommets $\{v_t : t = 1, 2, \ldots, T-1\}$ tels que $\delta^{\text{out}}(W_v^i) \cap A_T = \emptyset$ pour tout $i = 0, 1, \ldots, T$. On détermine, par énumération, l'ensemble W_v correspondant au sous-ensemble W_v^i, $i = 0, 1, \ldots, T$, de plus grand ratio. Le nouvel ensemble correspond au sous-ensemble de plus grand ratio parmi l'ensemble $\{W_v : v \in V \setminus \{v_0, o^k, d^k\}\}$.

Le calcul de ce nouveau sous-ensemble peut être réalisé en $O(n(\max\{\tilde{A}, p\}))$. En effet, pour chaque sommet $v \in V \setminus \{v_0, o^k, d^k\}$, la complexité du calcul de W_v est équivalente à celle de l'ajout des sommets $\{v_t : t = 1, 2, \ldots, T-1\}$ aux sous-ensembles V_1 et V_2 dans l'heuristique n-Partition.

Algorithme 3: Heuristique Max-Ratio-Cut

Entrée : - vecteur $\overline{y}$
- demande $k \in K$

Sortie : - Ensemble S de contraintes de coupes violées par $\overline{y}$

Début

$S = \emptyset$;
Choisir aléatoirement $W' \subseteq \bar{V}$ tel que $o_1^k, O^- \in W'$, $d_{T-1}^k, O^+ \in \bar{V} \setminus W'$ et $\delta^{\text{out}}(W') \cap A_T = \emptyset$;
Répéter
  $W = W'$;
  $W' =$ amélioration(W,k);
Jusqu'à $\rho(W') \leq \rho(W)$;
si *la contrainte de coupe associée à* V_1 *est violée par* $\overline{y}$ **alors**
  Ajouter dans S la contrainte de coupe associée à V_1;
retourner S;

Fin

Algorithme 4: Fonction amélioration

Entrée : - vecteur $\overline{y}$, ensemble $W \subseteq \bar{V}$
- demande $k \in K$

Sortie : - Ensemble $W' \subseteq \bar{V}$

Début

$W' = W$;
pour chaque *sommet* $v \in V \setminus \{v_0, o^k, d^k\}$ **faire**
  Soit W_v obtenu à partir de W en répartissant les sommets $\{v_t \mid t = 1, \ldots, T-1\}$ tel que $\delta^{\text{out}}(W) \cap A_T = \emptyset$ et $\rho(W_v)$ est maximum;
  si $\rho(W_v) > \rho(W')$ **alors**
    $W' = W_v$;
retourner W';

Fin

3.3.3 Séparation des contraintes étendues de capacité résiduelle

Nous présentons dans cette section la procédure de séparation que nous utilisons pour séparer les contraintes étendues de capacité résiduelle. Comme cet algorithme est basé sur celui séparant les contraintes de capacité résiduelles, nous présentons d'abord cette séparation.

Le problème de séparation pour les contraintes de capacité résiduelle peut être résolu en temps polynomial. Atamtürk et Rajan [10] ont montré que la séparation sur chaque arc peut être résolue en $\mathcal{O}(p)$. Pour un arc $a \in \tilde{A}$ donné, nous posons $R = \{k \in K \mid \overline{x}_a^k > q^k(\overline{y}_a - \lfloor \overline{y}_a \rfloor)\}$. Il ont montré que si une contrainte de type (3.11) associée à un arc a est violée par $(\overline{x}, \overline{y})$ pour un ensemble de demandes $K' \subseteq K$, il en existe alors une violée pour l'ensemble de demandes R. L'algorithme de séparation consiste alors à vérifier pour tout arc $a \in \tilde{A}$ si la contrainte associée est violée pour l'ensemble R. Ceci peut être fait en temps linéaire.

Le problème de séparation des contraintes étendues de capacité résiduelle peut être décomposé en $T-2$ sous-problèmes indépendants, chacun défini par une valeur $\phi \in \{1, \ldots, T-2\}$. La complexité de ce problème de séparation pour une valeur ϕ donnée est une question ouverte. Cependant, si l'ensemble d'arcs $X \subseteq A^\phi$ est fixé, le problème de séparation est le même que pour les contraintes de capacité résiduelle (3.11). La différence est que maintenannt, l'ensemble R est défini par $R = \{k \in K \mid \sum_{a \in X} \overline{x}_a^k > q^k(\sum_{a \in X} \overline{y}_a - \lfloor \sum_{a \in X} \overline{y}_a \rfloor)\}$. Donc, quand l'ensemble d'arcs X est fixé, les contraintes étendues de capacité résidelle sur les arcs associées à X peuvent être séparées en $\mathcal{O}(p)$.

Pour tout $\phi \in \{1, \ldots, T-2\}$, plutôt de séparer de manière heuristique toutes les contraintes étendues de capacité résiduelle, nous séparons de manière exacte les contraintes (3.12) associées à un arc de A^ϕ, deux arcs de A^ϕ, tous les arcs de A^ϕ sauf un, et tous les arcs de A^ϕ sauf deux. (On remarque que l'heuristique sépare aussi les contraintes de capacité résiduelle puisque ces dernières sont un cas particulier des contraintes étendues (3.12) lorsque $|X| = 1$.) La complexité de l'algorithme de séparation est en $\mathcal{O}(p \sum_{\phi=1}^{T-2} |A^\phi|(|A^\phi|+1))$ puisqu'il existe $\frac{|A^\phi|(|A^\phi|-1)}{2}$ paires d'arcs non ordonnées $a, a' \in A^\phi$ pour un $\phi \in \{1, \ldots, T-2\}$ donné.

3.4 Résultats expérimentaux

Cette section est consacrée aux résultats expérimentaux que nous avons obtenus lors de la résolution du problème de ramassage et livraison multi-véhicules asymétrique préemptif avec fractionnement basée sur les précédents développements théoriques. Nous commençons cette section en présentant l'algorithme de coupes et branchements que nous avons utilisé pour résoudre les deux formulations introduites en Section 3.1.

Commes les contraintes de coupes sont en nombre exponentiel, il est clair que ces dernières seront ajoutées au programme linéaire courant à l'aide des procédures de séparation définies précédemment. Cependant, afin de renforcer la relaxation linéaire initiale, il peut être intéressant d'ajouter un nombre polynomial de contraintes de coupes. Dans notre algorithme, nous ajoutons les contraintes de coupes associées aux sommets de V, c'est-à-dire,

$$y(\delta^{\text{out}}(\{u_1, u_2, \ldots, u_{T-1}\} \cup \{O^-\})) \geq \left\lceil \frac{q(K')}{Q} \right\rceil \tag{3.14}$$

s'il existe $K' \subseteq K$, $K' \neq \emptyset$, tel que $u = o^k$ pour tout $k \in K'$, et

$$y(\delta^{\text{out}}(V' \setminus \{u_1, u_2, \ldots, u_{T-1}\} \cup \{O^-\})) \geq \left\lceil \frac{q(K')}{Q} \right\rceil \tag{3.15}$$

s'il existe $K' \subseteq K$, $K' \neq \emptyset$, tel que $u = d^k$ pour tout $k \in K'$. Le programme linéaire initial pour la formulation multiflots correspond au programme linéaire

$$\min\{c^T y \mid (x, y) \in \mathbb{R}^{p|\tilde{A} \cup A_T|} \times \mathbb{R}^{|\bar{A}|} : (x, y) \text{ satisfait } (3.1) - (3.6), (3.14), (3.15)\}.$$

Pour la formulation métrique, le programme linéaire initial est correspond au programme linéaire

$$\min\{c^T y \mid y \in \mathbb{R}^{|\bar{A}|} : y \text{ satisfait } (3.1), (3.2), (3.6), (3.14), (3.15)\}.$$

Pour la formulation multiflots, les contraintes de coupes (3.10) sont séparées à chaque itération à l'aide des heuristiques 2 et 3. Si aucune contrainte de coupes violée n'est trouvée, nous séparons alors les contraintes étendues de capacité résiduelle (3.12) à l'aide de la procédure donnée dans la sous-section 3.3.3.

Pour la formulation métrique, la solution optimale de la relaxation doit être un vecteur entier satisfaisant toutes les contraintes métriques pour être réalisable pour le m-PRLAPF. À chaque itération, nous séparons d'abord les contraintes de coupes (3.10) à l'aide des trois heuristiques définies précédemment. Si aucune contrainte violée n'est

trouvée, nous séparons de manière exacte les contraintes métriques (3.8) et de manière heuristique les contraintes métriques arrondies à l'aide de la prodédure de séparation définie dans la sous-section 3.3.1.

Pour les deux formulations, lors de la séparation des contraintes de coupes, chaque fois que l'heuristique 3 est appliquée, dix sous-ensembles initiaux sont choisis aléatoirement. Parmi les dix contraintes trouvées, seule celle (si elle est violée par la solution courante) associée au sous-ensemble de plus grand ratio est ajoutée au programme linéaire courant. De plus, cette heuristique est appliquée systématiquement pour toute demande de K.

Pour nos tests, nous avons considéré des instances aléatoires. Les graphes représentant le réseau sont des graphes complets orientés provenant de la TSP Library [82]. Les coûts sur les arcs sont égaux aux arrondis supérieurs des distances Euclidiennes. Pour chaque graphe, trois ensembles de demandes de taille 5, 10 et 15 ont été générés aléatoirement de telle manière que le volume de chaque demande soit compris entre $Q/4$ et $3Q/4$. (Q correspond à la capacité des véhicules.) L'instance spécifie également la borne temporelle T et le nombre de véhicules. Afin de s'assurer que les instances sont réalisables, nous considérons que tous les arcs ont une durée de 1 et le nombre de véhicules disponibles est supérieur ou égal à $\frac{p}{\lfloor (T-1)/2 \rfloor}$. Cette valeur assure que l'instance est réalisable puisque chaque véhicule peut transporter au minimum $\lfloor \frac{T-1}{2} \rfloor$ demandes. En effet, chaque véhicule part du dépôt et va directement à l'origine de la première demande qu'il doit transporter. Pour chaque demande k des $\lfloor \frac{T-1}{2} \rfloor$ demandes, le véhicule la transporte sur l'arc (o^k, d^k). Une fois que le véhicule a atteint la destination de la demande k, il va directement à l'origine de la demande suivante, disons k', par l'arc $(d^k, o^{k'})$. Le véhicule termine sa route au dépôt (Nous remarquons que cette solution est réalisable car le volume de chaque demande est inférieur ou égal à la capacité des véhicules, le graphe est complet et les durées des arcs sont égales à 1.) Toutes les instances considérées sont donc réalisables.

Pour nos instances, nous avons fixé la borne temporelle à 7 et considéré deux valeurs pour les véhicules : $|F| = \frac{p}{\lfloor (T-1)/2 \rfloor}$ et $|F| = \frac{p}{\lfloor (T-1)/2 \rfloor} + 2$.

Dans les tables suivantes, les entrées sont :
- n : le nombre de sommets dans le graphe initial,
- p : le nombre de demandes,
- $|F|$: le nombre de véhicules,
- NC : le nombre de contraintes de coupes (3.10) générées,
- NMA : le nombre de contraintes métriques arrondies (3.13) générées,
- NM : le nombre de contraintes métriques (3.8) générées,

- NECR : le nombre de contraintes étendues de capacité résiduelle (3.12) générées,
- o/p : le nombre de problèmes résolus à l'optimum sur le nombre d'instances testées,
- Gap1 : l'erreur relative entre la meilleure borne supérieure (la valeur optimale si le problème a été résolu à l'optimum) et les bornes inférieures obtenues avant l'ajout de coupes pour renforcer la relaxation linéaire,
- Gap2 : l'erreur relative entre la meilleure borne supérieure et la borne inférieure obtenue avant branchement,
- CPU : Le temps de calcul en secondes.

Chaque ligne des tables rapporte les résultats moyens obtenus pour trois instances, toutes ayant le même nombre de sommets, de demandes et de véhicules. Les trois instances diffèrent par les coordonnées des sommets, et par les origines, les destinations et les volumes des demandes. Les deux formulations ont été testées sur les mêmes instances.

Chaque instance a d'abord été transformée en une instance dans le graphe auxiliaire comme décrit en Section 3.1. En conséquence, un graphe initial ayant n sommets donne lieu à un graphe auxiliaire avec $6n+2$ sommets et $5n^2+2n$ arcs. Cette transformation donne lieu à des programmes linéaires mixtes ayant $5n^2+2n$ variables pour la formulation métrique et $(p+1)\times 5n^2+2n$ variables pour la formulation multiflots. Par ailleurs, les programmes linéaires initiaux des formulations multiflots et métrique contiennent respectivement au plus $10n^2+np(6+5n)+5n+1$ et $5n^2+10n+1$ contraintes. (Le nombre de contraintes de coupes (3.14) et (3.15) est au plus de $2n$.)

La table 3.1 donne les résultats obtenus en utilisant la formulation multiflots. Nous remarquons que presque toutes les instances ayant 5 ou 10 demandes ont été résolues à l'optimalité à l'exception de 3 d'entre elles. Pour les instances avec 15 demandes, notre algorithme de coupes et branchements résout les instances jusqu'à 7 sommets. Cependant, pour des instances plus grandes, le problème apparaît plus dur à résoudre. En fait, seulement les deux tiers des demandes ayant 8 sommets (et 15 demandes) ont pu être résolues. De plus aucune instance avec 9 sommets ou plus n'a été résolue.

Nous pouvons également remarquer que notre algorithme de coupes et branchements génère un nombre important de contraintes de coupes (3.10) (jusqu'à 2125) et un nombre très important de contraintes étendues de capacité résiduelle (3.12) (jusqu'à 29175). Nous pouvons également remarquer que l'erreur relative Gap2 est faible pour la plupart des instances. De plus, elle est beaucoup plus petite que la valeur Gap1, ce qui montre que l'ajout des contraintes (3.10) et (3.12) renforce de manière significative la relaxation linéaire.

La table 3.2 rapporte les résultats obtenus en utilisant la formulation métrique. Nous remarquons que pour les instances ayant 9 sommets ou moins, tous les problèmes ont

n	p	$\|F\|$	NC	NECR	o/p	Gap1	Gap2	CPU
6	5	2	17.33	733.00	3/3	17.26	1.69	1.02
6	5	4	24.00	988.00	3/3	18.81	2.09	1.36
6	10	4	85.33	2261.66	3/3	13.82	1.64	42.86
6	10	6	96.00	3222.00	3/3	14.29	1.59	47.23
6	15	5	238.66	7103.00	3/3	13.35	4.16	362.11
6	15	7	242.00	7062.33	3/3	15.17	4.74	656.95
7	5	2	104.33	4977.66	3/3	33.24	20.72	64.05
7	5	4	38.00	2061.33	3/3	15.46	3.21	4.81
7	10	4	221.66	6582.00	3/3	17.17	5.44	208.24
7	10	6	84.00	3602.33	3/3	12.57	0.09	36.05
7	15	5	965.66	17662.66	3/3	17.59	8.75	5174.03
7	15	7	1292.66	17883.33	3/3	16.87	9.12	7313.39
8	5	2	103.00	3703.33	3/3	19.80	6.99	34.74
8	5	4	87.33	4960.66	3/3	20.03	6.49	33.66
8	10	4	344.33	9581.00	3/3	17.73	4.09	673.36
8	10	6	278.33	8175.66	3/3	16.75	3.83	647.17
8	15	5	1432.33	17653.33	2/3	18.24	5.79	9162.24
8	15	7	1447.00	20645.66	2/3	21.32	8.48	13587.84
9	5	2	39.00	5498.66	3/3	13.35	2.99	27.52
9	5	4	79.00	6973.33	3/3	16.93	4.05	62.16
9	10	4	550.66	12309.33	3/3	18.34	3.77	2179.85
9	10	6	785.33	17228.00	3/3	13.68	3.94	4952.95
9	15	5	1056.66	17126.66	0/3	32.71	25.16	18000.00
9	15	7	1447.66	19059.66	0/3	32.40	22.42	18000.00
10	5	2	146.33	8985.00	3/3	16.71	7.59	152.38
10	5	4	149.33	7220.66	3/3	16.77	7.96	562.95
10	10	4	2125.33	23081.00	1/3	26.10	11.99	13025.04
10	10	6	1571.66	29175.66	2/3	20.89	10.92	14093.96
10	15	5	1681.33	22241.00	0/3	38.57	28.37	18000.00
10	15	7	1907.33	27872.00	0/3	26.10	15.09	18000.00

TABLE 3.1 – Résultats obtenus avec la formulation multiflots

été résolus à l'optimum, à l'exception de deux d'entre eux. Nous pouvons également noter que pour toutes les instances ayant jusqu'à 10 sommets et 10 demandes, nous obtenons la solution optimale. Cependant, pour 10 sommets et 15 demandes, seulement une instance sur les six a été résolue. Ceci peut être expliqué par le fait que le problème devient plus dur quand la taille de l'instance augmente. De plus, il semble que le

n	p	$\|F\|$	NC	NMA	NM	o/p	Gap1	Gap2	CPU
6	5	2	17.66	0.00	0.00	3/3	17.26	1.82	0.24
6	5	4	34.33	0.66	0.00	3/3	18.81	2.76	0.96
6	10	4	76.00	4.33	0.00	3/3	13.82	1.70	3.95
6	10	6	106.00	10.33	0.00	3/3	14.29	2.02	4.88
6	15	5	213.33	29.33	0.00	3/3	13.35	4.39	26.74
6	15	7	271.33	60.00	0.00	3/3	15.17	5.21	51.99
7	5	2	304.33	1.66	0.00	3/3	33.24	20.89	36.39
7	5	4	61.33	0.00	0.00	3/3	15.46	3.21	2.85
7	10	4	284.66	23.00	0.00	3/3	17.17	5.49	29.78
7	10	6	60.00	4.00	0.00	3/3	12.57	0.19	1.43
7	15	5	1580.00	179.00	0.33	3/3	17.59	9.17	1620.19
7	15	7	1785.00	166.33	0.00	3/3	16.87	9.44	2176.62
8	5	2	121.33	1.00	0.00	3/3	19.80	7.00	6.90
8	5	4	187.66	0.66	0.00	3/3	20.03	6.50	16.69
8	10	4	440.00	21.33	0.00	3/3	17.73	4.13	120.67
8	10	6	436.33	31.66	0.00	3/3	16.75	4.43	129.39
8	15	5	3110.66	210.33	0.33	3/3	17.89	6.39	3073.67
8	15	7	1719.33	75.33	0.00	3/3	16.80	3.70	846.76
9	5	2	155.66	1.66	0.00	3/3	13.35	2.99	15.55
9	5	4	218.33	9.33	0.00	3/3	16.93	4.06	46.83
9	10	4	388.33	15.00	0.00	3/3	18.34	3.64	108.95
9	10	6	1072.00	31.00	0.00	3/3	13.68	3.94	2125.72
9	15	5	3488.00	364.00	0.66	1/3	16.90	8.24	14947.41
9	15	7	3098.33	226.33	1.66	3/3	14.60	2.43	5980.95
10	5	2	173.00	2.66	0.00	3/3	16.71	7.63	79.13
10	5	4	304.00	7.33	0.00	3/3	16.77	8.18	224.40
10	10	4	2582.66	50.33	0.00	3/3	20.38	4.85	6580.89
10	10	6	2314.66	71.00	0.00	3/3	16.52	5.94	6329.88
10	15	5	6105.66	151.33	0.00	1/3	20.83	8.04	14565.66
10	15	7	5038.66	228.00	0.33	0/3	26.30	16.01	18000.00

TABLE 3.2 – Résultats obtenus avec la formulation métrique

paramètre le plus décisif soit le nombre de demandes. En effet, comme on peut le voir sur la table 3.2, les instances avec 15 demandes nécessitent au minimum deux fois plus de temps de calcul que les instances ayant 10 demandes.

Nous pouvons remarquer que l'algorithme génère jusqu'à plusieurs centaines de contraintes métriques arrondies (3.13) alors qu'un nombre très faible de contraintes mé-

triques (3.8) est généré. Ceci s'explique par le fait que dans la plupart des cas, une contrainte métrique violée est étendue en une contrainte métrique arrondie en utilisant l'algorithme de séparation définie dans la section précédente. Cela montre que l'heuristique de séparation des contraintes métriques arrondies (3.13) est efficace. Nous remarquons également que l'algorithme génère un nombre important de contraintes de coupes (3.10). Finalement, comme pour la formulation multiflots, Gap2 est plus petit que Gap1. Les contraintes métriques arrondies (3.13) et les contraintes de coupes (3.10) sont donc très efficaces dans l'amélioration de la borne inférieure.

En comparant les résultats obtenus par les deux formulations, nous pouvons voir que plus d'instances ont été résolues à l'optimum avec la formulation métrique. De plus, le temps de calcul est environ 5 fois plus petit que celui obtenu pour la formulation multiflots. Cependant, cette dernière fournit une erreur relative (Gap2) plus faible. Cependant, pour quelques instances comme celles avec 9 sommets, 15 demandes et 7 véhicules, l'erreur relative Gap2 obtenue avec la formulation multiflots est plus grande que celle associée à la formulation métrique. Mais ceci peut s'expliquer par le fait que la borne supérieure obtenue avec la première formulation est assez grande.

3.5 Conclusion

Nous avons présenté deux formulations linéaires mixtes pour le problème m-PRLAPF et étudié un algorithme de coupes et branchements pour les résoudre. Nous avons identifié quelques contraintes valides. En particulier, nous avons introduit une nouvelle famille de contraintes qui généralisent les contraintes de capacité résiduelle. D'après les résultats expérimentaux, ces inégalités sont utiles pour résoudre la formulation multiflots.

Nos modèles peuvent facilement être étendus pour prendre en compte d'autres contraintes telles que les fenêtres de temps et les capacités de stockage à chaque noeud. Nous pouvons également modifier nos formulations pour prendre en compte une flotte hétérogène de véhicules. Pour cela, nous devons considérer des variables binaires pour chaque véhicule (plutôt que des variables entières associées à tous les véhicules). Il faudrait ajouter plusieurs dépôts et étendre les contraintes pour prendre en compte différentes capacités pour les véhicules.

Notre approche peut être vue comme un travail prospectif pour développer un algorithme de coupes et branchements efficace pour résoudre ce problème. Il serait intéressant de développer une heuristique afin d'obtenir rapidement une borne supérieure de

bonne qualité. Cette dernière permettrait d'améliorer les valeurs des erreurs relatives pour les instances qui ne sont pas résolues à l'optimum et d'accélérer l'algorithme par élagage de l'arbre de branchements. Il serait également avantageux d'identifier de nouvelles classes d'inégalités valides pour améliorer la borne inférieure. Par ailleurs, dû à l'utilisation d'un graphe auxiliaire, les nombres de variables et de contraintes des deux formulations augmentent très rapidement avec la taille de l'instance. Afin d'accélérer la résolution des relaxations linéaires, il serait judicieux d'essayer d'autres approches. En particulier, il serait intéressant de considérer un algorithme de génération de colonnes basé sur une formulation arc-chemin. Une autre possibilité serait d'appliquer une décomposition de Benders pour la formulation métrique. Finalement, il pourrait être avantageux d'essayer de trouver une autre modélisation n'utilisant pas de graphe auxiliaire. Cette question est l'objet du chapitre suivant pour un cas particulier du problème lorsqu'un seul véhicule est disponible et le fractionnement des demandes interdit.

Chapitre 4

Le problème de ramassage et livraison mono-véhicule préemptif asymétrique

D'après le chapitre précédent, nous avons pu observer que le problème de ramassage et livraison asymétrique multi-véhicules préemptif avec fractionnement de la demande est un problème très complexe à résoudre de manière exacte. Malgré les améliorations possibles que l'on peut tenter d'apporter à notre algorithme, voire l'utilisation d'un autre schéma de résolution tel que la génération de colonnes ou certaines décompositions, il semble que l'on ne peut espérer réellement arriver à des résultats intéressants à l'heure actuelle. Ce problème combine plusieurs difficultés telles que le transport avec préemption ou le fractionnement de la demande qui, prises séparément, suffisent à rendre les problèmes de ramassage et livraison très complexes. Il apparaît alors naturel de se focaliser sur une seule de ces difficultés, en l'occurence le transport avec préemption, pour tenter de comprendre clairement les difficultés sous-jacentes induites par la prise en compte de cette possibilité. De plus, si nous arrivons à développer un algorithme efficace pour résoudre le problème de ramassage et livraison préemptif, nous pouvons espérer que cet algorithme pourra être utilisé par la suite comme sous-routine dans le schéma de résolution du m-PRLAPF.

On s'intéresse ainsi au *Problème de Ramassage et Livraison mono-véhicule Préemptif Asymétrique (1-PRLPA)*. Bien que ce dernier soit un cas particulier du m-PRLPF, il n'en reste pas moins un problème très complexe. Dans le 1-PRLPA, on limite les circuits du véhicule à ceux traversant chaque arc du graphe au maximum une fois. Cette limitation s'explique par la difficulté engendrée lorsque le véhicule traverse plusieurs fois un même arc en transportant des demandes. Il faut alors en effet savoir sur quel passage du véhicule sur l'arc chaque demande est transportée. Par ailleurs, cette li-

mitation n'est pas importante puisqu'il est possible de la contourner en ajoutant des arcs multiples - le véhicule passant alors une fois par chaque arc multiple. Finalement, on montre également, dans le chapitre 6, que lorsque le véhicule ne transporte qu'une seule demande à la fois, cette limitation, sous certaines hypothèses, n'en est pas une puisque, dans ce cas, il existe toujours une solution optimale où chaque arc est traversé au maximum une fois par le véhicule.

Dans ce chapitre, nous étudions tout d'abord la structure des solutions du problème. Nous montrons que dans le cas général, les solutions du 1-PRLPA doivent être décrites par l'ensemble des arcs traversés par le véhicule, l'ordre dans lequel ces derniers sont traversés, et les arcs des chemins des demandes. Nous prouvons aussi que ces informations sont nécessaires pour reconnaître en temps polynomial si une solution est réalisable. En d'autres termes, nous montrons que si une de ces informations n'est pas disponible, le problème de reconnaissance de la réalisabilité d'une solution est NP-complet. Nous nous intéressons également au problème lorsque le véhicule ne peut transporter qu'une seule demande à la fois. Cette variante, appelée *Problème de Ramassage et Livraison mono-véhicule Préemptif Asymétrique unitaire (1-PRLPA unitaire)*, reste NP-difficile. Cependant, nous montrons qu'il n'est pas nécessaire - contrairement au cas général - d'avoir l'ordre sur les arcs du véhicule pour décrire une solution. En effet, même sans cette information, il est possible de reconnaître en temps polynomial si une solution est réalisable. Autrement dit, il est dans ce cas possible, à partir des ensembles d'arcs associés au véhicule et aux demandes, de savoir en temps polynomial s'il existe un ordre sur les arcs traversés par le véhicule conforme au circuit du véhicule et aux chemins des demandes. Nous terminons ce chapitre par la présentation de deux formulations entières : l'une qui est valide dans le cas unitaire et l'autre dans le cas général. La formulation unitaire sera ensuite étudiée dans les chapitres suivants et un algorithme de coupes et branchements sera développé pour résoudre le 1-PRLPA unitaire.

4.1 Description d'une solution et complexité

L'information choisie pour décrire les solutions d'un problème d'optimisation combinatoire est directement liée à l'efficacité d'un algorithme de résolution par approche polyédrale puisqu'elle influe sur les variables utilisées. Or, comme mentionné par Queyranne et Schultz [81], "le succès de cette approche dépend fortement du choix des variables qui est typiquement la première question que l'on doit se poser lorsque l'on veut formuler un problème".

L'ensemble des variables considéré dans une formulation linéaire en nombres entiers

correspond en fait à une représentation d'une certaine information qui peut induire plusieurs solutions pas nécessairement réalisables. Une *représentation manipulable* doit contenir suffisamment d'information pour permettre de tester en temps polynomial si cette dernière correspond ou non à une solution réalisable. Une *représentation minimale* (au sens de l'inclusion) est une représentation manipulable dans laquelle aucune information ne peut être enlevée sans perdre la possibilité de savoir en temps polynomial si l'information restante correspond à une solution réalisable. Une représentation minimale implique un nombre minimal de solutions. En conséquence, travailler avec ce nombre réduit de solutions possibles dans des méthodes basées sur l'énumération implicite (i.e., méthode de séparation et évaluation, méthode de coupes et branchements) permet de limiter l'explosion combinatoire.

Dans cette partie, on s'intéresse à déterminer la représentation minimale des solutions du 1-PRLPA. Dans le chapitre 2, on a donné les conditions nécessaires et suffisantes pour qu'une solution du m-PRLR soit réalisable. On redonne maintenant ces conditions de manière plus formelle en l'adaptant au cas mono-véhicule. Pour cela, on introduit de nouvelles notations. Soit $P = (a_1, a_2, \ldots, a_b)$, $b \geq 1$ un chemin (ou circuit) d'un graphe. On dit qu'un arc a précède un arc a' dans P, et l'on note $a \prec_P a'$, si $a, a' \in P$ et a apparaît avant a' dans la séquence P. Si X est un ensemble de chemins, on dit de deux arcs a et a' que a précède a' dans X, et l'on note $a \prec_X a'$, s'il existe un chemin P de X tel que $a \prec_P a'$.

Une solution du problème consiste en un circuit (non élémentaire) du graphe D, noté C, correspondant au circuit du véhicule et un ensemble de p chemins élémentaires de D, chaque chemin correspondant au trajet d'une demande. On note par L_k le chemin de o^k à d^k correspondant au trajet de la demande $k \in K$. Une solution (C, L) avec $L = \{L_1, L_2, \ldots, L_p\}$ est réalisable si et seulement si :

i - le circuit C commence au dépôt v_0,

ii - les chemins L_k, $k \in K$, utilisent uniquement des arcs de C,

iii - la somme des volumes des demandes passant par chaque arc ne dépasse pas Q,

iv - $a \prec_L a'$ implique que $a \prec_C a'$ pour tout couple d'arcs $a \neq a'$ du graphe D.

La première condition permet de s'assurer que le véhicule passe par le dépôt v_0. Les conditions (ii) et (iii) obligent le véhicule à transporter les demandes et à ne pas dépasser sa capacité. (Comme le véhicule traverse chaque arc au maximum une fois, il est évident que la capacité sur les arcs traversés par le véhicule est égale à Q.) Finalement, la condition (iv) impose que l'ordre induit par le circuit du véhicule soit conforme aux chemins des demandes.

Une représentation manipulable pour le 1-PRLPA correspond alors à donner les chemins des demandes et le circuit du véhicule. Comme nous l'avons déjà mentionné,

le trajet d'une demande $k \in K$ correspond à un chemin élémentaire de o^k à d^k dans le graphe D. L'ensemble d'arcs est alors suffisant pour représenter ce chemin puisque la séquence peut être trivialement retrouvée à partir de l'ensemble. Comme le circuit du véhicule n'est pas élémentaire, ce dernier est alors représenté par l'ensemble des arcs traversés et l'ordre dans lequel ils sont traversés. Il est clair qu'avec ces informations, on peut vérifier en temps polynomial si toute solution vérifie les quatre conditions précédemment définies.

La question logique est maintenant de savoir si une telle représentation manipulable est minimale. En d'autres termes, peut-on supprimer une information parmi l'ensemble des arcs traversés par les demandes, l'ensemble des arcs du circuit du véhicule et l'ordre sur les arcs du circuit du véhicule, et garder tout de même une représentation manipulable ?

Il est clair que l'on ne peut pas supprimer l'ensemble des arcs traversés par le véhicule sans supprimer l'ordre sur ces arcs. De plus, on ne peut pas décrire les solutions uniquement à l'aide des chemins des demandes. Deux représentations contenant moins d'information sont alors envisageables. Dans la première, les solutions sont décrites uniquement par la séquence du véhicule, c'est-à-dire l'ensemble des arcs traversés par le véhicule et l'ordre dans lequel ils sont traversés. Dans la deuxième représentation, les solutions sont décrites par les ensembles d'arcs associés au circuit du véhicule et aux chemins des demandes. Dans cette section, on montre que ces deux représentations ne sont pas manipulables pour le cas général du 1-PRLPA, ce qui implique que la représentation manipulable définie précédemment est minimale. Pour cela, on prouve que les problèmes consistant à vérifier si une solution décrite par chacune de ces représentations est réalisable pour le 1-PRLPA sont NP-complets. On montre cependant que lorsque le véhicule ne peut transporter qu'une demande à la fois, la deuxième représentation est une représentation manipulable. Ceci signifie que nous pouvons vérifier en temps polynomial si une solution décrite par les arcs du circuit du véhicule et les chemins des demandes est réalisable.

Pour les deux problèmes de satisfiabilité, on note par A' l'ensemble des arcs de A traversés par le véhicule. Cet ensemble induit un graphe orienté Eulérien que l'on note $D' = (V', A')$, où $V' = V(A')$. Comme v_0 est le sommet de départ (et d'arrivée) du trajet du véhicule, il est clair que v_0 appartient à V'. De même, V' contient les origines et destinations des demandes de K. De plus, comme le véhicule passe exactement une fois par chaque arc de A', Q unités de demandes peuvent être transportées au maximum sur chaque arc de A'.

4.1.1 Le problème de satisfiabilité des demandes

On prouve dans cette partie que les solutions du 1-PRLPA ne peuvent être uniquement décrites par la séquence du véhicule. En effet, dans ce cas, on montre que vérifier si une solution est réalisable est un problème NP-complet. Ce problème, appelé *Problème de Satisfiabilité des Demandes (PSD)*, tente de répondre à la question suivante. Soit C un circuit Eulérien de D'. Existe-t-il un ensemble $L = \{L_k : k \in K\}$ de chemins de demandes de D' tel que pour chaque demande $k \in K$, les arcs du $o^k d^k$-chemin L_k sont traversés dans le même ordre que dans C, et pour chaque arc $a \in A'$, le volume total des demandes traversant a ne dépasse pas la capacité du véhicule Q ? On montre que le PSD est NP-complet, même lorsque le véhicule ne transporte qu'une demande à la fois (*i.e.*, les chemins des demandes sont arc-disjoints), chaque sommet excepté le dépôt est incident à au plus une demande, le dépôt est incident à aucune demande, aucun sommet de D' n'apparait plus de deux fois dans C et v_0 apparait uniquement dans C comme sommet de départ et d'arrivée.

On prouve que le PSD est NP-complet par réduction du problème des chemins arc-disjoints dans les graphes quasi-topologiques. Avant de définir clairement ce nouveau problème NP-complet, on introduit la définition des graphes quasi-topologiques. Soit $G = (V_G, A_G)$ un graphe orienté simple acyclique avec $V_G = \{v_1, v_2, \ldots, v_n\}$. On suppose que le graphe G contient un circuit hamiltonien dont les arcs sont ceux de l'ensemble $A_H = \{(v_i, v_{i+1}) : i = 1, 2, \ldots, n-1\}$. Les arcs de $A_J = A_G \setminus A_H$ sont appelés *arcs de saut*. On remarque que comme G est simple et acyclique, tout arc de A_j est de la forme (v_i, v_j) avec $1 \leq i < j \leq n$ et $j - i > 1$. Le graphe G est *quasi-topologique* si tous les degrés entrants et sortants sont inférieurs ou égaux à deux, et chaque sommet est incident à au plus trois arcs. Par définition de G, chaque sommet de V_G est incident à au plus un arc de saut. Soit $S = \{(s_i, t_i) : i = 1, 2, \ldots, r\}$ un ensemble de couples de sommets distincts tel que chaque sommet de G appartient à au plus un couple, et les sommets de chaque couple de S sont incidents à aucun arc de saut. (On a alors $\deg^{\text{in}}(v) \leq 1$ et $\deg^{\text{out}}(v) \leq 1$ pour tout $v \in \{s_i, t_i : i = 1, \ldots, r\}$.) Le *Problème des Chemins Arc-Disjoints dans les graphes Quasi-Topologiques (PCADQT)* consiste à vérifier s'il existe r chemins élémentaires arc-disjoints $X_1, X_2, \ldots, X_r$ tel que X_i est un $s_i t_i$-chemin de G pour tout $i = 1, 2, \ldots, r$. Dans le lemme suivant, on prouve que le PCADQT est NP-complet.

Lemme 4.1 *Le problème des chemins arc-disjoints dans les graphes quasi-topologiques est NP-complet.*

Preuve. On prouve que le problème des chemins arc-disjoints dans les graphes quasi-

topologiques est NP-complet par réduction du problème des chemins arc-disjoints dans les graphes acycliques. Ce dernier peut être défini de la manière suivante. Le *Problème des Chemins Arc-Disjoints dans les graphes Acycliques (PCADA)*, consiste, étant donnés un graphe orienté acyclique (non nécessairement simple) $G' = (V_{G'}, A_{G'})$ et un ensemble S' de r' couples de sommets distincts (s'_i, t'_i), à déterminer s'il existe r' chemins élémentaires arc-disjoints $X'_1, X'_2, \ldots, X'_{r'}$ tel que X'_i est un $s'_i t'_i$-chemin de G' pour tout $i = 1, \ldots, r'$. Ce problème a été montré NP-complet par Vygen [98]. Un exemple d'une instance du PCADA est donné dans la figure 4.1, où l'information relative à S' est donnée entre parenthèses.

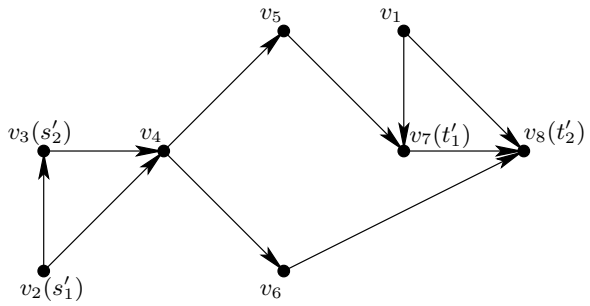

FIGURE 4.1 – Une instance de PCADA

On explique maintenant la construction d'une instance (G, S) du PCADQT à partir d'une instance (G', S') du PCADA. Tout d'abord, pour $i = 1, 2, \ldots, r'$, on crée deux nouveaux sommets s_i et t_i et deux nouveaux arcs (s_i, s'_i) et (t'_i, t_i). Soit $S = \{(s_i, t_i) : i = 1, 2, \ldots, r'\}$. On remarque qu'aucun sommet n'appartient à deux couples de S, ce qui peut être le cas avec S'. Considérons le graphe $G^* = (V_{G^*}, A_{G^*})$ où $V_{G^*} = V_{G'} \cup \{s_i, t_i : i = 1, 2, \ldots, r'\}$ et $A_{G^*} = A_{G'} \cup \{(s_i, s'_i), (t'_i, t_i) : i = 1, 2, \ldots, r'\}$. Comme G^* est acyclique, on peut calculer un ordre topologique T de G^*. De plus, T est choisi de telle manière que le sommet s'_i (respectivement t_i) suive immédiatement le sommet s_i (respectivement t'_i), pour tout $i = 1, 2, \ldots, r'$. Soit $\tilde{A}$ l'ensemble des arcs (u, v) tel que v est immédiatement après u dans T et l'arc (u, v) n'appartient pas à A_{G^*}. Il est clair que le graphe orienté $\tilde{G}$ induit par $A_{\tilde{G}} = A_{G^*} \cup \tilde{A}$ contient un chemin hamiltonien traversant les sommets de V_{G^*} dans le même ordre que dans T. Notons par A'_H les arcs de ce chemin hamiltonien. Le graphe ainsi obtenu à partir de celui de la figure 4.1 par cette construction est donné dans la figure 4.2, où les arcs pointillés représentent l'ensemble d'arcs $\tilde{A}$.

À ce stade, le graphe orienté peut ne pas être quasi-topologique puisque les conditions de degré peuvent ne pas être satisfaites ou le graphe peut contenir des arcs

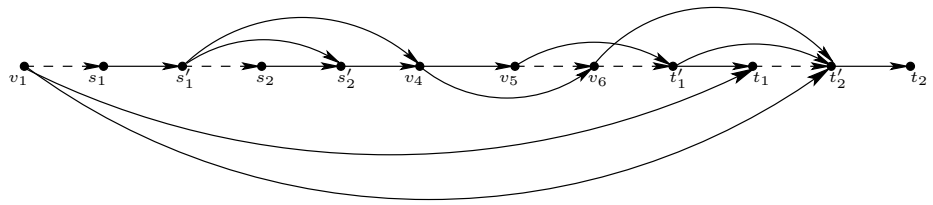

FIGURE 4.2 – Graphe orienté $\tilde{G}$

multiples. Soit $\tilde{W}$ le sous-ensemble de sommets v de V_{G^*} tel que $r_v = \deg^{\text{in}}_{\tilde{G}}(v) > 2$, $s_v = \deg^{\text{out}}_{\tilde{G}}(v) > 2$, ou $d_v = \deg_{\tilde{G}}(v) > 3$. Il est clair que $\tilde{W}$ ne contient aucun sommet d'un couple de S car $\deg^{\text{in}}(v) \leq 1$ et $\deg^{\text{out}}(v) \leq 1$ pour tout $v \in \{s_i, t_i : i = 1, \ldots, r'\}$. Soit v un sommet de $\tilde{W}$. Les arcs entrants (respectivement sortants) de v sont ordonnés de telle manière que s'il existe un arc de A'_H dans $\delta^{\text{in}}(v)$ (respectivement $\delta^{\text{out}}(v)$), alors celui-ci est le premier (respectivement dernier) arc dans l'ordre. On considère $z_v = r_v(s_v+1)+s_v(r_v+1)$ nouveaux sommets $v^1, v^2, \ldots, v^{z_v}$. On remplace alors dans $\tilde{G}$ le sommet v par le chemin $((v^1, v^2), (v^2, v^3), \ldots, (v^{z_v-1}, v^{z_v}))$ de la manière suivante : le i^{e} arc de $\delta^{\text{in}}_{\tilde{G}}(v)$ a $v^{i(s_v+1)-s_v}$ pour extrémité terminale $i = 1, 2, \ldots, r_v$; le i^{e} arc de $\delta^{\text{out}}_{\tilde{G}}(v)$ a $v^{r_v(s_v+1)+i(r_v+1)}$ comme extrémité initiale pour $i = 1, 2, \ldots, s_v$. Pour $i = 1, 2, \ldots, r_v$, notons par I^i_v la séquence des (s_v+1) sommets consécutifs partant de $v^{i(s_v+1)-s_v}$ (correspondant à l'extrémité terminale du i^{e} arc entrant de v dans $\tilde{G}$). L'ensemble I^i_v est associé au i^{e} arc entrant. De manière similaire, on définit l'ensemble O^i_v de sommets en considérant les $(r_v + 1)$ sommets consécutifs finissant par $v^{r_v(s_v+1)+i(r_v+1)}$ pour $i = 1, 2, \ldots, s_v$. L'ensemble O^i_v est associé au i^{e} arc sortant. Par ailleurs, si $s_v \neq 0$ et $r_v \neq 0$, on ajoute également les arcs $(v^{i(s_v+1)+j+1}, v^{r_v(s_v+1)+j(r_v+1)-r_v+i})$ pour $i = 0, 1, \ldots, r_v - 1$ et pour $j = 1, 2, \ldots, s_v$. Ce dispositif que nous utilisons pour remplacer un sommet v de $\tilde{W}$ est illustré dans la figure 4.3 où $\delta^{\text{in}}_{\tilde{G}}(v) = \{a_1, a_2\}$ et $\delta^{\text{out}}_{\tilde{G}}(v) = \{a_3, a_4, a_5\}$. (Les arcs de A'_H sont les arcs a_1 et a_5.)

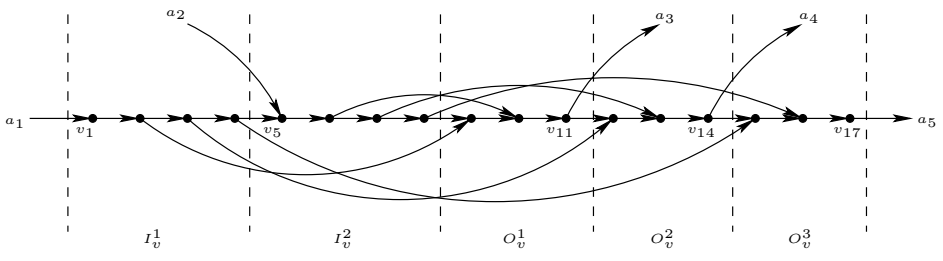

FIGURE 4.3 – Dispositif de remplacement d'un sommet de $\tilde{W}$

Le graphe ainsi obtenu, disons $\hat{G}$, contient un chemin hamiltonien. Notons par A^*_H les arcs de ce chemin. À ce stade, le graphe $\hat{G}$ peut toujours ne pas être simple si le

graphe $\tilde{G}$ contenait des arcs multiples qui n'étaient pas incidents à $\tilde{W}$. Cependant, par définition $\tilde{W}$, étant donné deux sommets u et v de $\hat{G}$, il existe au plus deux arcs multiples de u à v et l'un des deux appartient à A_H^*. On subdivise alors l'arc multiple (u, v) appartenant à A_H^*, *i.e.*, on ajoute un sommet w et on remplace l'arc (u, v) de A_H^* par les arcs (u, w) et (w, v). En répétant cette opération pour tous les arcs parallèles de A_H^*, on obtient un graphe simple.

Le dernier dispositif que nous utilisons consiste à remplacer chaque arc $a = (u, v)$ de $\tilde{A}$ par un chemin $((u, s_a), (s_a, t_a), (t_a, v))$, où s_a et t_a sont deux nouveaux sommets. De plus, le couple (s_a, t_a) est ajouté à S. A la fin de cette procédure de remplacement, l'ensemble S est alors composé de $r = r' + |\tilde{A}|$ couples de sommets. Soit G le graphe orienté ainsi obtenu. Il est clair que tous les sommets de G satisfont les conditions de degré d'un graphe quasi-topologique. De plus, le graphe est simple donc G est un graphe quasi-topologique. Comme aucun sommet de G n'appartient à deux couples de S et $\deg^{\text{in}}(v) \leq 1$ et $\deg^{\text{out}}(v) \leq 1$ pour tout $v \in \{s_i, t_i : i = 1, \ldots, r\}$, le graphe orienté G et l'ensemble S correspondent à une instance du PCADQT. Comme trouver un ordre topologique dans un graphe acyclique peut être fait en temps polynomial, la construction de G et S à partir de G' et S' peut être faite en temps polynomial.

On montre maintenant que l'instance (G', S') du PCADA admet une solution réalisable si et seulement si l'instance (G, S) du PCADQT en admet une. On considère d'abord une solution de l'instance (G, S) du PCADQT, c'est-à-dire, r chemins arc-disjoints $X_1, X_2, \ldots, X_r$. Sans perte de généralité, un chemin X_i pour $i = 1, 2, \ldots, r'$ est associé à un couple de S'. Soit $X'_1, X'_2, \ldots, X'_{r'}$ la restriction de $X_1, X_2, \ldots, X_{r'}$ sur le graphe orienté G'. À cause de la définition des couples (s_a, t_a) avec $a \in \tilde{A}$, X'_i correspond à un chemin de G' de s'_i à t'_i. Comme les chemins $X_1, X_2, \ldots, X_{r'}$ sont arc-disjoints, les chemins $X'_1, X'_2, \ldots, X'_{r'}$ le sont aussi.

Pour prouver la réciproque, on considère une solution de l'instance (G', S') du PCADA donnée par r' chemins arc-disjoints $X'_1, X'_2, \ldots, X'_{r'}$. Pour $i = 1, 2, \ldots, r'$, chaque chemin X'_i peut être trivialement étendu à un chemin $\tilde{X}_i$ de $\tilde{G}$ en concaténant l'arc (s_i, s'_i), le chemin X'_i et l'arc (t'_i, t_i). Les chemins $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_{r'}$ sont clairement arc-disjoints. De plus, comme ni (s_i, s'_i) ni (t'_i, t_i) n'appartient à $\tilde{A}$, le chemin $\tilde{X}_i$ ne contient aucun arc de $\tilde{A}$, pour $i = 1, 2, \ldots, r'$. Soient v un sommet de $\tilde{W}$ et q le nombre de chemins de $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_{r'}$ traversant v. Sans perte de généralité, on suppose que ces chemins sont les q premiers. On remarque que $1 \leq q \leq \min\{r_v, s_v\}$. Considérons n'importe quel i de $\{1, 2, \ldots, q\}$, et notons par a_i (respectivement b_i) l'arc de $\tilde{X}_i$ entrant dans (respectivement sortant de) v. Soient $I_v^{i'}$ et $O_v^{i''}$ les sous-ensembles de sommets associés respectivement à a_i et b_i, comme définis précédemment dans le dispositif de remplacement. On remarque que dans G, le sommet $v^{i'(s_v+1)-s_v}$ est l'extrémité terminale de a_i,

le sommet $v^{r_v(s_v+1)+i''(r_v+1)}$ est l'extrémité initiale de b_i. D'après la définition du dispositif de remplacement, il est clair qu'il existe un unique chemin M_i de $v^{i'(s_v+1)-s_v}$ à $v^{r_v(s_v+1)+i''(r_v+1)}$ dans le sous-graphe de G induit par $I_v^{i'} \cup O_v^{i''}$. On construit maintenant le chemin $\bar{X}_i$ en insérant M_i entre les arcs a_i et b_i dans $\tilde{X}_i$. On remarque qu'à partir de deux chemins arc-disjoints $\tilde{X}_j$ et $\tilde{X}_{j'}$ avec $1 \leq j, j' \leq q$, $j \neq j'$, on obtient deux chemins arc-disjoints M_j et $M_{j'}$. Comme les chemins $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_q$ sont arc-disjoints, les chemins $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_q$ le sont aussi. En réïtérant ce processus d'insertion local pour chaque sommet de $\tilde{W}$, on obtient r' chemins $X_1, X_2, \ldots, X_{r'}$ de G qui sont arc-disjoints. Pour finir la preuve, on considère le chemin composé des arcs (s_a, t_a) pour tout couple de S associé à un arc a de $\tilde{A}$. De cette façon, on obtient $r = r' + |\tilde{A}|$ chemins arc-disjoints dans G. □

En utilisant une réduction à partir du problème des chemins arc-disjoints dans les graphes quasi-topologiques, nous prouvons dans le théorème suivant que le problème de satisfiabilité des demandes est NP-complet.

Théorème 4.2 *Le problème de satisfiabilité des demandes est NP-complet, même si $Q = 1$, aucun sommet n'apparait plus de deux fois dans le circuit Eulérien C et v_0 apparait uniquement comme sommet de départ et d'arrivée de C. De plus, le problème reste NP-complet dans le cas où l'ensemble des demandes K est tel que chaque sommet de $V' \setminus \{v_0\}$ est incident à au plus une demande, le dépôt v_0 étant incident à aucune demande.*

Preuve. Considérons une instance du PCADQT donnée par un graphe orienté quasi-topologique $G = (V_G, A_G)$ et un ensemble $S = \{(s_i, t_i) : i = 1, 2, \ldots, r\}$ de couples de sommets disjoints de G tel que chaque sommet de G appartient à au plus un couple et, $\deg^{\text{in}}(v) \leq 1$ et $\deg^{\text{out}}(v) \leq 1$ pour tout $v \in \{s_i, t_i : i = 1, \ldots, r\}$.

Pour obtenir le graphe orienté D' de l'instance du PSD, on contracte d'abord chaque arc de saut de G. (Les arcs de saut sont ceux n'appartenant pas au chemin hamiltonien de G.) On ajoute ensuite un nouveau sommet v_0 et deux nouveaux arcs (v_0, v_1) et (v_n, v_0). Comme G est quasi-topologique et la contraction des arcs de saut entraîne la formation de circuits, le graphe orienté D' ainsi obtenu est clairement Eulérien. Le circuit Eulérien C de D' est construit comme suit. On considère la séquence $((v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n), (v_n, v_0))$ où $((v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n))$ correspond au chemin hamiltonien de G. Pour tout arc (v_i, v_j) de A_J, on identifie dans C les sommets v_i et v_j. Il est évident que C est un circuit Eulérien de D'. L'ensemble K des demandes est obtenu à partir de S en associant une demande k_i au couple (s_i, t_i) pour

tout $i = 1, 2, \ldots, r$, tel que o^{k_i} (respectivement d^{k_i}) est le sommet de D' correspondant à s_i (respectivement t_i) et $q^{k_i} = 1$. Il est clair que les sommets o^{k_i} et d^{k_i} sont distincts car on a $\deg^{\text{in}}(v) \leq 1$ et $\deg^{\text{out}}(v) \leq 1$ pour tout sommet de $\{s_i, t_i : i = 1, 2, \ldots, r\}$. Chaque sommet de D' est alors incident à au plus une demande et le dépôt v_0 est incident à aucune demande. De plus, on fixe la capacité Q du véhicule à un. Il est clair que cette transformation est polynomiale.

On prouve maintenant que le PCADQT a une solution réalisable si et seulement si le PSD en possède une. Considérons une solution réalisable du PCADQT définie par r chemins arc-disjoints $X_1, X_2, \ldots, X_r$ de G. On construit une solution pour le PSD de la manière suivante. Pour $i = 1, 2, \ldots, r$, le chemin L_i de D' associé avec la demande k_i est obtenu de X_i en supprimant tous les arcs de saut. Comme les chemins $X_1, X_2, \ldots, X_r$ sont arc-disjoints, la capacité du véhicule est respectée pour tout arc de D'. De plus, les arcs de L_i sont traversés dans le même ordre que dans C, pour $i = 1, 2, \ldots, p$, car G est quasi-topologique et C est construit à partir du chemin hamiltonien de G. En conséquence, $L_1, L_2, \ldots, L_p$ est une solution réalisable pour le PSD.

Pour prouver la réciproque, on commence avec une solution $L_1, L_2, \ldots, L_p$ réalisable pour le PSD. Comme $Q = 1$, les chemins L_i, $i = 1, 2, \ldots, p$ de D' sont arc-disjoints. Soit $i \in \{1, 2, \ldots, p\}$. Considérons la restriction X'_i de L_i sur G. Clairement, tous les arcs de X'_i appartiennent à A_H. Pour transformer X'_i en un $s_i t_i$-chemin, on ajoute les arcs de saut de A_J pour connecter les différents sous-chemins X'_i. On remarque que tout arc de saut ajouté correspond à un rechargement de k_i par rapport à C. Comme $X'_1, X'_2, \ldots, X'_p$ sont arc-disjoints, un arc de saut peut être ajouté au maximum une fois. Les $s_i t_i$-chemins obtenus pour $i = 1, 2, \ldots, p$ forment alors une solution réalisable pour le PCADQT. □

Le théorème précédent implique que toute représentation manipulable du 1-PRLPA contient l'information relative aux chemins des demandes. En d'autres termes, toute solution du 1-PRLPA doit être décrite par l'ensemble des arcs traversés par chaque demande.

4.1.2 Le problème de satisfiabilité de la séquence du véhicule

Nous tentons maintenant de décrire les solutions du problème de ramassage et livraison préemptif asymétrique mono-véhicule à l'aide des ensembles des arcs traversés par le véhicule et les demandes. Nous montrons que dans le cas général, une telle représentation n'est pas manipulable, c'est-à-dire qu'elle ne permet pas de savoir en temps

polynomial si une solution du 1-PRLPA est réalisable. Nous prouvons cependant à la fin de cette partie que si le véhicule transporte une seule demande à la fois, cette représentation est alors manipulable.

Le problème consistant à vérifier si une solution du 1-PRLPA décrite par les ensembles d'arcs est réalisable est appelé *Problème de Satisfiabilité de la Séquence du Véhicule (PSSV)*. Ce dernier consiste, étant donnés les arcs traversés par le véhicule et les chemins des demandes, à vérifier s'il existe un ordre sur les arcs traversés par le véhicule conforme aux chemins des demandes et correspondant à un circuit Eulérien commençant au dépôt. Autrement dit, dans le circuit Eulérien induit par l'ordre, les arcs doivent être traversés dans le même ordre que dans les chemins des demandes.

On présente maintenant le *Problème du Circuit Eulérien avec Contraintes de Précédence induites par des Chemins (PCECPC)*. Une instance du PCECPC est représentée par un triplet (H, u_0, P) où $H = (U, B)$ est un graphe orienté Eulérien, u_0 un sommet particulier de U et $P = \{P_1, P_2, \ldots, P_t\}$ un ensemble de chemins élémentaires de H. Le problème consiste à vérifier s'il existe un circuit Eulérien C de H commençant en u_0 et conforme aux chemins de P, *i.e.*, C satisfait $a \prec_C a'$ si $a \prec_P a'$ pour tout couple d'arcs $a \neq a'$ de B. Ces conditions sont appelées *contraintes de précédence induites par les chemins de* P. Si C est conforme à P, on dit aussi que C respecte P.

Le PSSV n'est rien d'autre que le PCECPC avec l'instance (D', v_0, L). En effet, on peut supposer, sans perte de généralité, que (D', v_0, L) vérifie les conditions (ii) et (iii). De plus, les conditions (i) et (iv) sont vérifiées si et seulement si le PCECPC admet une solution pour l'instance (D', v_0, L). Dans ce cas, le circuit Eulérien commence en v_0 et l'ordre des arcs du circuit Eulérien trouvé est conforme aux chemins des demandes.

On étudie dans cette partie la complexité du PCECPC. Les résultats de complexité trouvés permettront de savoir si le problème de satisfiabilité de la séquence du véhicule peut être résolu en temps polynomial et donc de savoir si la représentation décrivant les solutions du 1-PRLPA à l'aide des ensembles des arcs traversés par le véhicules et les différentes demandes est manipulable.

4.1.2.1 NP-complétude

On prouve que le problème du circuit Eulérien avec contraintes de précédence induites par des chemins est NP-complet par réduction à partir du *Problème du Circuit Hamiltonien dans les graphes orientés de degré deux (2-PCH)* qui est défini comme suit. Étant donné un graphe orienté $D_H = (V_H, A_H)$ tel que tous ses sommets sont de degré

entrant et sortant égaux à deux (i.e., $\deg^{\text{in}}(v) = \deg^{\text{out}}(v) = 2$ pour tout $v \in V_H$), peut-on trouver dans D_H un circuit hamiltonien, c'est-à-dire, un circuit traversant chaque sommet de V_H exactement une fois ? Le 2-PCH est un problème NP-complet [77]. En fait, la preuve donnée par Plesnik porte sur les graphes orientés planaires pour lesquels les degrés entrants et sortants sont bornés par deux pour chaque sommet. Cependant, en considérant quelques arcs additionnels, on peut très facilement étendre ce résultat aux graphes (avec arcs multiples) avec des degrés entrants et sortants égaux à deux.

Soit $V_H = \{v_1, v_2, \ldots, v_n\}$ avec $n \geq 2$. Notre réduction consiste à construire à partir de D_H un graphe orienté $H = (U, B)$ ayant $(4n + 2)$ sommets et $(10n + 2)$ arcs, et à définir un sommet de départ u_0 et un ensemble P de $(2n + 1)$ chemins élémentaires de H. On montre ensuite que H contient un circuit Eulérien commençant en u_0 et conforme à P si et seulement si D_H contient un circuit hamiltonien.

On construit le graphe H de la manière suivante. Au sommet v_1 de V_H, on associe dans H six sommets $v_1^1, v_1^2, v_1^3, v_1^4, w_1, w_2$, et les dix arcs suivants :

(i) $(v_1^1, v_1^3), (v_1^3, v_1^2), (v_1^2, v_1^4)$,

(ii) $(v_1^1, w_1), (w_1, w_2), (w_2, v_1^2)$,

(iii) $(v_1^4, w_1), (w_1, v_1^2)$,

(iv) $(v_1^3, w_2), (w_2, v_1^3)$.

Soit B_1 l'ensemble de ces dix arcs (voir figure 4.4).

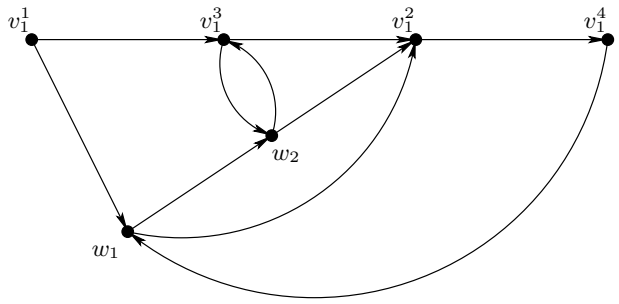

FIGURE 4.4 – Transformation du sommet v_1

Pour tout $i \in \{2, 3, \ldots, n\}$, on associe, à v_i, quatre sommets $v_i^1, v_i^2, v_i^3, v_i^4$, avec les huit arcs suivants

(v) $(v_i^1, v_i^3), (v_i^3, v_i^2), (v_i^2, v_i^4)$,

(vi) (v_i^1, v_i^2),

(vii) $(v_i^4, w_1), (w_1, v_i^2)$,

(viii) $(v_i^3, w_2), (w_2, v_i^3)$.

Soit B_i l'ensemble des huit arcs, pour tout $i = 2, 3, \ldots, n$ (voir figure 4.5). Remarquons que la seule différence entre les transformations des sommets v_1 et v_i, $i = 2, 3, \ldots, n$, est que v_i^1 et v_i^2 sont adjacents, alors que v_1^1 et v_1^2 ne le sont pas (*i.e.*, seuls les ensembles (ii) et (vi) diffèrent).

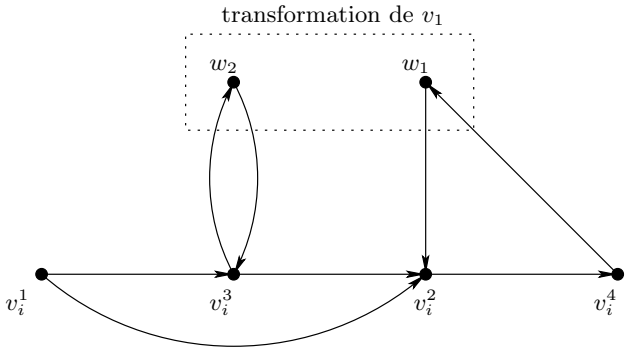

FIGURE 4.5 – Transformation du sommet v_i, $i = 2, 3, \ldots, n$

Soit

$$U = \{v_i^j : i = 1, 2, \ldots, n \text{ et } j = 1, 2, 3, 4\} \cup \{w_1, w_2\}.$$

À ce stade, tous les sommets de U ont leur degré entrant égal à leur degré sortant, excepté pour les sommets v_i^1 et v_i^2, pour tout $i = 1, 2, , \ldots, n$, dont la différence entre leur degrés entrant et sortant est égale à deux. On considère maintenant les arcs de D_H de la manière suivante. Pour tout arc (v_i, v_j) dans A_H, on associe l'arc (v_i^2, v_j^1) dans H. (voir figure 4.6.)

Soit

$$B = \left(\bigcup_{i=1}^{n} B_i \right) \cup \{(v_i^2, v_j^1) : (v_i, v_j) \in A_H\}.$$

Le graphe orienté $H = (U, B)$ est clairement faiblement connexe. De plus, étant donné que tous les sommets de D_H ont un degré entrant et un degré sortant égaux à deux, les sommets v_i^1 et v_i^2, pour $i = 1, 2, , \ldots, n$, ont maintenant leur degré entrant égaux à leur degré sortant. Par conséquent, le graphe H est Eulérien.
Pour obtenir une instance du PCECPC, on doit définir un sommet de départ u_0 de H (qui sera le sommet de départ du circuit Eulérien) ainsi qu'un ensemble P de chemins élémentaires de H. Soit $u_0 = v_1^2$. Pour tout $i = 1, 2, \ldots, n$, soient Q_i et R_i les chemins de H définis par

– $Q_i = ((v_i^4, w_1), (w_1, w_2), (w_2, v_i^3), (v_i^3, v_i^2))$,

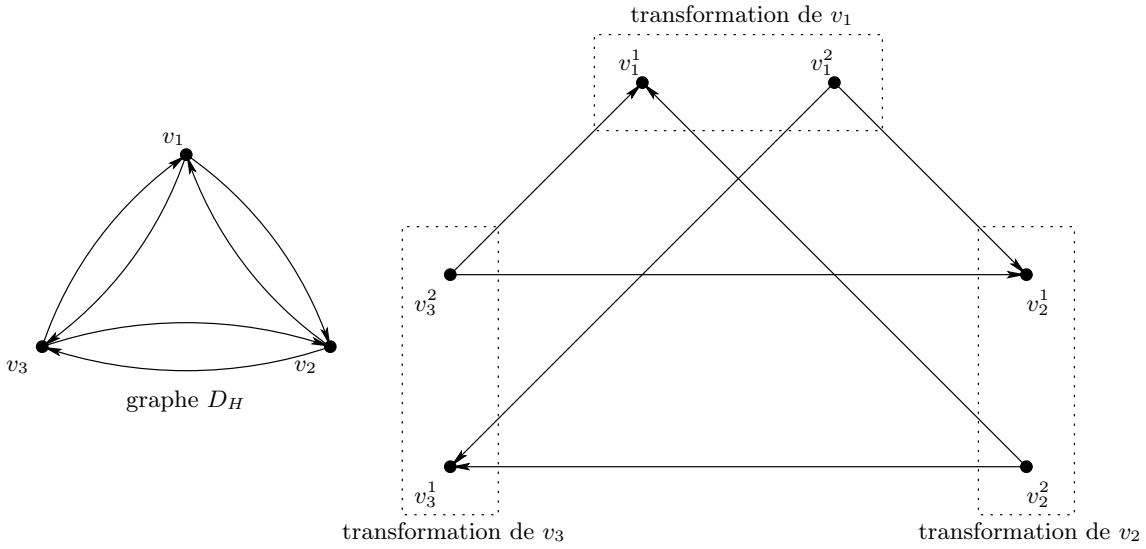

FIGURE 4.6 – Transformation des arcs de A_H

– $R_i = ((v_i^4, w_1), (w_1, v_i^2))$.

L'ensemble P de chemins élémentaires qui doit être respecté par le circuit Eulérien de H est le suivant

$$P = \{Q_i : i = 1, 2, \ldots, n\} \cup \{R_i : i = 1, 2, \ldots, n\} \cup \{\big((v_1^1, w_1), (w_1, w_2), (w_2, v_1^2)\big)\}.$$

Il est facile de voir que l'instance (H, u_0, P) peut être construite, à partir de D_H, en temps polynomial.

Lemme 4.3 *Si D_H a un circuit hamiltonien, alors H a un circuit Eulérien, commençant en u_0, respectant les contraintes de précédence spécifiées par P.*

Preuve. Soit C_H un circuit hamiltonien de D_H. Sans perte de généralité, on suppose que $C_H = ((v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n), (v_n, v_1))$. Notre preuve consiste à construire, en plusieurs étapes, un circuit Eulérien de H.

La première séquence d'arcs que l'on considère est obtenue en substituant l'arc (v_i, v_{i+1}) de C_H par le chemin

$$\big((v_i^2, v_i^4), (v_i^4, w_1), (w_1, v_i^2), (v_i^2, v_{i+1}^1), (v_{i+1}^1, v_{i+1}^2)\big)$$

pour tout $i = 1, 2, \ldots, n-1$, et l'arc (v_n, v_1) par le chemin

$$\big((v_n^2, v_n^4), (v_n^4, w_1), (w_1, v_n^2), (v_n^2, v_1^1), (v_1^1, w_1), (w_1, w_2)\big)\,.$$

Soit C_1 cette séquence d'arcs qui correspond à un chemin commençant en v_1^2, puisque C_H est un circuit élémentaire de D_H. Notons qu'aucun des arcs (w_2, v_i^3), (v_i^3, w_2), pour tout $i = 1, 2, \ldots, n$, ni l'arc (w_2, v_1^2) apparaît dans C_1. Considérons maintenant le graphe $H^* = (U^*, B^*)$ induit par les arcs restants de H, *i.e.*, par $B \setminus C_1$. On a $U^* = U \setminus (\{v_i^4 : i = 1, 2, \ldots, n\} \cup \{w_1\})$ et

$$\deg^{\text{in}}_{H^*}(v) - \deg^{\text{out}}_{H^*}(v) = \begin{cases} 1 & \text{si } v = v_1^2, \\ -1 & \text{si } v = w_2, \\ 0 & \text{si } v \in U^* \setminus \{v_1^2, w_2\}. \end{cases}$$

De plus, H^* est faiblement connexe puisqu'il contient les arcs (w_2, v_i^3), (v_i^1, v_i^3) et (v_i^3, v_i^2) pour tout $i = 1, 2, \ldots, n$, et qu'ils couvrent tous les sommets dans U^*. Supprimer l'arc (w_2, v_1^2) ne déconnecte pas le graphe ; ceci implique que $H^* - (w_2, v_1^2) = (U^*, B^* \setminus \{(w_2, v_1^2)\})$ est un graphe Eulérien. Par ailleurs, le graphe $H^* - w_2 = H^*[U^* \setminus \{w_2\}]$ peut ne pas être faiblement connexe mais chaque sommet de $U^* \setminus \{w_2\}$ a son degré entrant égal à son degré sortant. Soient $H_1^*, H_2^*, \ldots, H_s^*$, $s \geq 1$, les composantes fortement connexes de $H^* - w_2$. Ces dernières sont clairement des graphes Eulériens. Comme $H^* - (w_2, v_1^2)$ est faiblement connexe et tout arc incident à w_2 dans ce graphe appartient à l'ensemble $\{(w_2, v_i^3), (v_i^3, w_2) : i = 1, 2, \ldots, n\}$, alors pour tout $l = 1, 2, \ldots, s$, il doit exister $i_l \in \{1, 2, \ldots, n\}$ tel que $v_{i_l}^3$ est un sommet de H_l^*. Considérons n'importe quelle composante fortement connexe H_l^*, $l = 1, 2, \ldots, s$. Tout circuit Eulérien C_l^* de H_l^* commençant en $v_{i_l}^3$ peut être transformé en un circuit C_l de H^* commençant en w_2 de la manière suivante. Le premier arc de C_l est $(w_2, v_{i_l}^3)$. Tous les arcs de C_l^* sont ensuite ajoutés séquentiellement à C_l. Cependant, si l'extrémité terminale de l'arc ajouté est un sommet v_i^3, $i \in \{1, 2, \ldots, n\}$, alors le circuit $((v_i^3, w_2), (w_2, v_i^3))$ est ajouté à C_l avant de passer à l'arc suivant de C_l^*. Une fois que l'on a considéré tous les arcs de C_l^*, on complète C_l en ajoutant l'arc $(v_{i_l}^3, w_2)$. Comme tout circuit C_l, $l = 1, 2, \ldots, s$, commence au sommet w_2, la concaténation $(C_1, C_2, \ldots, C_s)$ de ces circuits forment un circuit Eulérien C_2 de $H^* - (w_2, v_1^2)$.
Soit C la concanétation de C_1, C_2 et de l'arc (w_2, v_1^2). Notons que C est composé de tous les arcs de B. De plus, étant donné que C_1 commence au sommet v_1^2 et termine au sommet w_2 qui est le sommet initial de C_2, C est un circuit Eulérien de H commençant en v_1^2. Pour tout $i = 1, 2, \ldots, n$, les chemins $((v_i^4, w_1), (w_1, w_2))$ et $((w_2, v_i^3), (v_i^3, w_2))$ sont respectivment des sous-chemins de C_1 et C_2. Ceci implique que C respecte les chemins Q_i pour tout $i = 1, 2, \ldots, n$. De plus, comme les chemins R_i, $i = 1, 2, \ldots, n$, et le chemin $((v_1^1, w_1), (w_1, w_2))$ sont des sous-chemins de C_1 et l'arc (w_2, v_1^2) appartient à C_2, il en résulte que C respecte tous les chemins de P, ce qui termine la preuve. □

Afin de prouver la contraposée du lemme 4.3, on doit prouver le résultat technique suivant.

Proposition 4.4 *Si le graphe H a un circuit Eulérien C, commençant en v_1^2, qui respecte les contraintes de précédence induites par P, alors $((v_i^2, v_i^4), (v_i^4, w_1), (w_1, v_i^2))$ est un sous-circuit de C pour tout $i = 1, 2, \ldots, n$.*

Preuve. Pour tout $i = 1, 2, \ldots, n$, le sommet v_i^4 a exactement un arc entrant et un arc sortant dans H. Par conséquent, $((v_i^2, v_i^4), (v_i^4, w_1))$ est un sous-chemin de C. De plus, comme C respecte tous les chemins R_i de P, on a

$$(v_i^2, v_i^4) \prec_C (v_i^4, w_1) \prec_C (w_1, v_i^2) \quad \text{pour tout } i = 1, 2, \ldots, n.$$

Supposons qu'il existe $j \in \{1, 2, \ldots, n\}$ tel que $\big((v_j^2, v_j^4), (v_j^4, w_1), (w_1, v_j^2)\big)$ n'est pas un sous-circuit C, *i.e.*, un arc $a_1 \in \delta^{\text{out}}(w_1) \backslash \{(w_1, v_j^2)\}$ suit (v_j^4, w_1) dans C, ce qui implique que $a_1 \prec_C (w_1, v_j^2)$. Soit $I \subset \{1, 2, \ldots, n\}$ les indices i des sommets v_i^4 qui apparaissent dans C avant v_j^4. Sans perte de généralité, on suppose que $((v_i^2, v_i^4), (v_i^4, w_1), (w_1, v_i^2))$ est un sous-circuit de C pour tout $i \in I$. On a alors

$$(w_1, v_i^2) \prec_C (v_j^4, w_1) \prec_C (v_{i'}^4, w_1) \quad \text{pour tout } i \in I \text{ et } i' \notin I \cup \{j\}. \tag{4.1}$$

Il est clair que a_1 ne peut pas être un des arcs (w_1, v_i^2) avec $i \in I$. Si $a_1 = (w_1, v_{i'}^2)$ avec $i' \notin I \cup \{j\}$, alors, à cause du chemin $R_{i'}$ (respecté par C), on doit avoir $(v_{i'}^4, w_1) \prec_C (w_1, v_{i'}^2)$, *i.e.*, $(v_{i'}^4, w_1) \prec_C (v_j^4, w_1)$, ce qui contredit (4.1). Par conséquent, a_1 doit être l'arc (w_1, w_2). Comme C respecte tous les chemins Q_i de P, on doit avoir $(v_i^4, w_1) \prec_C (w_1, w_2)$ pour tout $i = 1, 2, \ldots, n$. Comme C est un circuit Eulérien et $\delta^{\text{out}}(w_1) = \{(w_1, v_i^2) : i = 1, 2, \ldots, n\} \cup \{(w_1, w_2)\}$, il en résulte que $(w_1, v_i^2) \prec_C (w_1, w_2)$ pour tout $i = 1, 2, \ldots, n$. On déduit alors que $(w_1, v_j^2) \prec_C a_1 = (w_1, w_2)$, ce qui contredit la définition de a_1. L'arc suivant (v_j^4, w_1) dans C est donc l'arc (w_1, v_j^2). □

Nous pouvons maintenant montrer la contraposée du lemme 4.3.

Lemme 4.5 *Si H a un circuit Eulérien, commençant en v_1^2, qui respecte les contraintes de précédence induites par les chemins de P, alors D_H a un circuit hamiltonien.*

Preuve. Soit C un tel circuit Eulérien de H. Comme C respecte tous les chemins de P, l'arc (w_1, w_2) apparaît dans C après tous les autres arcs entrants de w_1 dans H, *i.e.*,

$$a \prec_C (w_1, w_2) \quad \text{pour tout } a \in \delta^{\text{in}}(w_1),$$

et avant tous les arcs sortants de w_2 dans H, *i.e.*,

$$(w_1, w_2) \prec_C a \quad \text{pour tout } a \in \delta^{\text{out}}(w_2).$$

De plus, comme $\deg^{\text{out}}(v) = \deg^{\text{in}}(v)$ pour tout $v \in U$, on déduit que

$$a \prec_C (w_1, w_2) \quad \text{pour tout } a \in \delta^{\text{out}}(w_1) \setminus \{(w_1, w_2)\}, \tag{4.2}$$

et

$$(w_1, w_2) \prec_C a \quad \text{pour tout } a \in \delta^{\text{in}}(w_2) \setminus \{(w_1, w_2)\}. \tag{4.3}$$

Par ailleurs, en considérant les chemins Q_i de P, on obtient également

$$(w_1, w_2) \prec_C (v_i^3, v_i^2) \quad \text{pour tout } i = 1, 2, \ldots, n,$$

qui, combiné avec (4.3), signifie que tous les sommets v_i^3, $i = 1, 2, \ldots, n$, apparaissent après (w_1, w_2) dans C.
Soit $\overline{C}$ le chemin obtenu à partir de C en considérant uniquement les arcs de H précédant (w_1, w_2) dans C. On montre maintenant que $\overline{C}$ contient un circuit hamiltonien dans D_H, après une série de suppressions d'arcs et de contractions de sommets (voir figure 4.7). Notons que $\overline{C}$ commence en v_1^2, termine en w_1 et ne contient aucun sommet

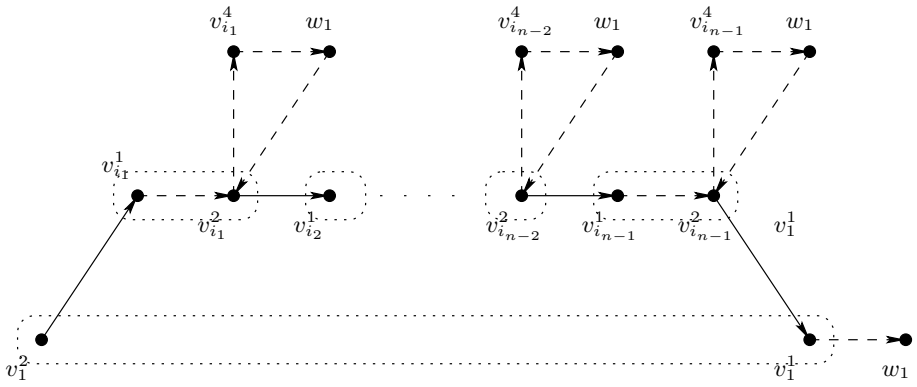

FIGURE 4.7 – Circuit $\overline{C}$; les traits pleins sont les arcs du circuit hamiltonien ; les tirets sont les arcs supprimés ; chaque paire de sommets entourée de pointillés est contractée.

de $\{v_i^3 : i = 1, 2, \ldots, n\}$. Comme C respecte le chemin $((v_1^1, w_1), (w_1, w_2), (w_2, v_1^2))$, on sait que le sommet v_1^2 apparaît exactement une fois dans $\overline{C}$, comme sommet initial, et le sommet v_1^1 une seule fois dans $\overline{C}$. De plus, le sommet v_1^1 a deux arcs sortants, correspondant à (v_1^1, v_1^3) et (v_1^1, w_1), et le sommet v_1^3 n'apparaît pas dans $\overline{C}$. Par conséquent, le sommet v_1^1 est l'avant dernier sommet dans $\overline{C}$, *i.e.*, $\overline{C}$ termine avec l'arc (v_1^1, w_1). D'après (4.2), l'arc (v_i^4, w_1) apparaît dans $\overline{C}$ pour tout $i = 1, 2, \ldots, n$. La proposition 4.4 implique alors que $((v_i^2, v_i^4), (v_i^4, w_1), (w_1, v_i^2))$ est un sous-circuit C_i de $\overline{C}$. Supprimer les circuits $C_1, C_2, \ldots, C_n$ de $\overline{C}$ et l'arc (v_1^1, w_1) donne lieu à un circuit $\tilde{C}$ commençant

en v_1^2, terminant en v_1^1, et contenant tous les sommets de $\{v_i^2 : i = 1, 2, \ldots, n\}$. Comme $\tilde{C}$ ne contient aucun sommet de $\{v_i^3 : i = 1, 2, \ldots, n\}$, on a alors

$$\delta_H^{\text{in}}(v_i^2) \cap \tilde{C} = \{(v_i^1, v_i^2)\} \quad \text{pour tout } i = 2, 3, \ldots, n.$$

et $\tilde{C}$ contient alors tous les sommets de $\{v_i^1, v_i^2 : i = 1, 2, \ldots, n\}$ exactement une fois. Par conséquent, $\tilde{C}$ une séquence alternée d'arcs de $\{(v_i^2, v_j^2) : (v_i, v_j) \in A_H\}$ et de $\{(v_i^1, v_i^2) : i = 2, 3, \ldots, n\}$ telle que chaque arc des deux ensembles apparaît une fois, le sommet initial est v_1^2 et le sommet terminal est v_1^1. Contracter les sommets v_i^1 et v_i^2 en v_i, pour tout $i = 1, 2, \ldots, n$, transforme $\tilde{C}$ en un circuit hamiltonien de D_H. □

Proposition 4.6 *Le problème du circuit Eulérien avec contraintes de précédence induites par des chemins est NP-complet.*

Preuve. Clairement, le problème est dans NP. De plus, la contruction d'une instance du 2-PCH en une instance du PCECPC peut être faite en temps polynomial. La NP-complétude du problème du circuit Eulérien avec contraintes de précédence induites par des chemins provient directement des lemmes 4.3 et 4.5. □

Dans la preuve que l'on vient de donner, aucune hypothèse n'a été faite pour le graphe H. On montre dans le corollaire suivant que si ce dernier est simple, le résultat de complexité reste vrai.

Corollaire 4.7 *Le problème du circuit Eulérien avec contraintes de précédence induites par des chemins reste NP-complet si H est un graphe orienté simple.*

Preuve. Considérons une instance du PCECPC définie par un graphe Eulérien $H = (U, B)$ ayant des arcs multiples, un sommet de départ u_0 et un ensemble P de chemins élémentaires de H. Soit $H' = (U', B')$ le graphe simple obtenu à partir de H en remplaçant séquentiellement chaque arc multiple (u, v) par les deux arcs (u, w) et (w, v) où w est un nouveau sommet de degrés entrant et sortant égaux à un. Les mêmes substitutions sont appliquées aux chemins de P. Notons P' ce nouvel ensemble de chemins dans H'. La construction de H' et P' peut évidemment être faite en temps polynomial. De plus, il est facile de voir que les deux instances du PCECPC (i.e., celle définie par H, u_0 et P, et celle définie par H', u_0 et P') sont équivalentes. □

Le reste de cette partie est consacrée à prouver que le PCECPC reste NP-complet même si l'ensemble P vérifie certaines restrictions. Ces restrictions sont en rapport avec le nombre de successeurs et de prédecesseurs qu'un arc de B a dans P, et avec la longueur des chemins dans P. On introduit maintenant quelques définitions et notations. Soit X un ensemble de chemins élémentaires. Si deux arcs a et a' sont adjacents dans un chemin de X, a est alors un *prédécesseur* de a' dans X et a' un *successeur* de a dans X. On note, pour tout arc $a \in B$, par $\mathcal{S}_X(a)$ (respectivement $\mathcal{P}_X(a)$) l'ensemble des successeurs (respectivement prédécesseurs) distincts de a dans X. Soit $\mathcal{S}_3(P)$ l'ensemble des arcs de B ayant au moins trois successeurs dans P, *i.e.*,

$$\mathcal{S}_3(P) = \{a \in B : |\mathcal{S}_P(a)| \geq 3\}.$$

On note $\sigma_3(P)$ la cardinalité de la famille $(\mathcal{S}_P(a) : a \in \mathcal{S}_3(P))$, *i.e.*,

$$\sigma_3(P) = \sum_{a \in \mathcal{S}_3(P)} |\mathcal{S}_P(a)|.$$

De manière similaire pour les arcs ayant au moins trois prédécesseurs, on définit

$$\mathcal{P}_3(P) = \{a \in B : |\mathcal{P}_P(a)| \geq 3\},$$

et

$$\pi_3(P) = \sum_{a \in \mathcal{P}_3(P)} |\mathcal{P}_P(a)|.$$

Lemme 4.8 *Considérons un graphe Eulérien $H = (U, B)$, un sommet u_0 of H et un ensemble P de chemins élémentaires de H. Si $\sigma_3(P) \geq 3$, l'instance (H, u_0, P) du PCECPC peut réduite en temps polynomial en une instance (H', u_0, P') du PCECPC dans laquelle $\sigma_3(P') \leq \sigma_3(P) - 1$ et $\pi_3(P') \leq \pi_3(P)$.*

Preuve. Comme $\sigma_3(P) \geq 3$, il est clair que $\mathcal{S}_3(P)$ est non vide. Soit $a_1 = (v_1, v_2)$ un arc de $\mathcal{S}_3(P)$, et considérons un arc $a_2 = (v_2, v_3)$ dans $\mathcal{S}_P(a_1)$.
Tout d'abord, on construit le graphe $H' = (U', B')$ à partir de H en

(ix) remplaçant l'arc a_1 par les deux arcs (v_1, w_1) et (w_1, v_2), où w_1 est un nouveau sommet,

(x) remplaçant l'arc a_2 par les deux arcs (v_2, w_2) et (w_2, v_3), où w_2 est un nouveau sommet,

(xi) en ajoutant les trois arcs (w_1, w_2), (w_2, w_3) et (w_3, w_1), où w_3 est un nouveau sommet.

Soit B_w l'ensemble de ces nouveaux arcs, *i.e.*,

$$B_w = \{(v_1, w_1), (w_1, v_2), (v_2, w_2), (w_2, v_3), (w_1, w_2), (w_2, w_3), (w_3, w_1)\}.$$

On a alors $U' = U \cup \{w_1, w_2, w_3\}$ et $B' = (B \setminus \{a_1, a_2\}) \cup B_w$ (voir figure 4.8). Notons que chaque sommet de $\{v_i, w_i : i = 1, 2, 3\}$ a son degré entrant égal à son degré sortant dans H'. Comme H est un graphe Eulérien, alors H' l'est aussi.

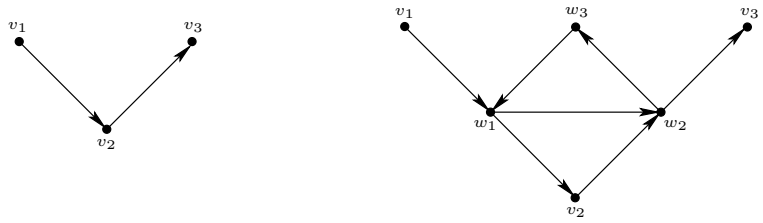

FIGURE 4.8 – Ensemble d'arcs B_w de H'

On définit ensuite les chemins élémentaires de H' qui compose l'ensemble P'. On remarque, qu'étant donné que P est seulement composé de chemins élémentaires de H, les arcs a_1 et a_2 ne peuvent apparaître dans un même chemin sans que a_2 suive a_1. Pour chaque chemin Q de K, on associe dans P' un chemin de la manière suivante

(xii) on garde Q inchangé si Q ne contient ni a_1 ni a_2,

(xiii) on remplace a_1 par le sous-chemin $((v_1, w_1), (w_1, v_2))$ si Q contient a_1 mais pas a_2,

(xiv) on remplace a_2 par le sous-chemin $((v_2, w_2), (w_2, v_3))$ si Q contient a_2 mais pas a_1,

(xv) on remplace le sous-chemin (a_1, a_2) par le sous-chemin $((v_1, w_1), (w_1, w_2), (w_2, v_3))$ si Q contient les deux arcs a_1 et a_2,

On ajoute également à P' les deux chemins élémentaires $((w_1, w_2), (w_2, w_3))$ et $((v_2, w_2), (w_2, v_3))$. L'ensemble P' est donc composé de $|P| + 2$ chemins élémentaires de H'. Il est clair que la transformation de (H, u_0, P) en (H', u_0, P') peut être faite en temps polynomial.

Comme les sommets w_1, w_2 et w_3 ont leurs degrés entrant et sortant bornés par deux, aucun des arcs (v_1, w_1), (v_2, w_2), (w_1, w_2), (w_2, w_3) et (w_3, w_1) appartient à $\mathcal{S}_3(P')$. De même, les arcs (w_1, v_2), (w_2, v_3), (w_1, w_2), (w_2, w_3) et (w_3, w_1) n'appartiennent pas à $\mathcal{P}_3(P')$. Par ailleurs, d'après la construction de H', on a

$$|\mathcal{S}_{P'}(a)| = |\mathcal{S}_P(a)| \quad \text{pour tout } a \in B' \setminus B_w, \tag{4.4}$$

$$|\mathcal{P}_{P'}(a)| = |\mathcal{P}_P(a)| \quad \text{pour tout } a \in B' \setminus B_w. \tag{4.5}$$

De plus, à cause de (xiii) et (xv), on obtient facilement

$$|\mathcal{S}_{P'}((w_1, v_2))| = |\mathcal{S}_P(a_1)| - 1, \tag{4.6}$$
$$|\mathcal{P}_{P'}((v_1, w_1))| = |\mathcal{P}_P(a_1)|. \tag{4.7}$$

En utilisant (xiv) et (xv), on déduit également que

$$|\mathcal{S}_{P'}((w_2, v_3))| = |\mathcal{S}_P(a_2)|, \tag{4.8}$$
$$|\mathcal{P}_{P'}((v_2, w_2))| = |\mathcal{P}_P(a_2)| - 1. \tag{4.9}$$

D'après (4.4), on a alors $\mathcal{S}_3(P') \cap (B' \setminus B_w) = \mathcal{S}_3(P) \cap (B \setminus \{a_1, a_2\})$, et d'après (4.5), $\mathcal{P}_3(P') \cap (B' \setminus B_w) = \mathcal{P}_3(P) \cap (B \setminus \{a_1, a_2\})$. Par conséquent, pour comparer $\sigma_3(P')$ avec $\sigma_3(P)$, et $\pi_3(P')$ avec $\pi_3(P)$, on doit seulement considérer les arcs a_1 et a_2 pour P et les arcs de B_w pour P'. Supposons que $a_2 \in \mathcal{S}_3(P)$; le raisonnement est similaire si $a_2 \notin \mathcal{P}_3(P)$. D'après (4.8), on a $(w_2, v_3) \in \mathcal{S}_3(P')$. D'après (4.4), (4.6) et (4.8), on obtient alors

$$\begin{aligned}
\sigma_3(P') &= \sum_{\substack{a \in \mathcal{S}_3(P') \\ a \notin B_w}} |\mathcal{S}_{P'}(a)| + \sum_{\substack{a \in \mathcal{S}_3(P') \\ a \in \{(w_1,v_2),(w_2,v_3)\}}} |\mathcal{S}_{P'}(a)| \\
&\leq \sum_{\substack{a \in \mathcal{S}_3(P) \\ a \notin \{a_1,a_2\}}} |\mathcal{S}_P(a)| + |\mathcal{S}_P(a_1)| - 1 + |\mathcal{S}_P(a_2)| \\
&= \sigma_3(P) - 1,
\end{aligned}$$

l'inégalité venant du fait que $(w_1, v_2) \notin \mathcal{S}_3(P')$ si $|\mathcal{S}_P(a_1)| = 3$. En utilisant (4.5), (4.7) et (4.9), on prouve de manière similaire que $\pi_3(P') \leq \pi_3(P)$.
On doit maintenant prouver que (H, u_0, P) a une solution réalisable si et seulement si (H', u_0, P') en a également une. Considérons un circuit Eulérien C de H respectant P. Comme $a_2 \in \mathcal{S}_P(a_1)$, on doit avoir $a_1 \prec_C a_2$. Soit C' le circuit Eulérien de H' obtenu à partir de C en remplaçant respectivement les arcs a_1 et a_2 par les chemins $((v_1, w_1), (w_1, w_2), (w_2, w_3), (w_3, w_1), (w_1, v_2))$ et $((v_2, w_2), (w_2, v_3))$. Comme C respecte P, il est facile de voir que C' respecte tous les chemins élémentaires de P' générés par (xii), (xiii) et (xiv). De plus, d'après les substitutions de a_1 et a_2, C' respecte les chemins $((w_1, w_2), (w_2, w_3))$ et $((v_2, w_2), (w_2, v_3))$. Puisque $a_1 \prec_C a_2$, on a également $(w_1, w_2) \prec_{C'} (w_2, v_3)$, ce qui implique que C' respecte tous les chemins de P' générés par (xv). Par conséquent, C' est une solution réalisable de l'instance (H', u_0, P') du PCECPC.
Considérons maintenant un circuit Eulérien $\overline{C}'$ de H' respectant P'. Si $((w_1, w_2), (w_2, v_3))$ est un sous-chemin de $\overline{C}'$, alors, en considérant les chemins de P' générés par (xiii) et le chemin $((w_1, w_2), (w_2, w_3))$, on obtient

$$(v_2, w_2) \prec_{\overline{C}'} (w_1, w_2) \prec_{\overline{C}'} (w_2, v_3) \prec_{\overline{C}'} (w_2, w_3).$$

Or, ceci contredit le fait que $\overline{C}'$ est un circuit Eulérien. Par conséquent, les chemins $((w_1, w_2), (w_2, w_3))$ et $((v_2, w_2), (w_2, v_3))$ sont des sous-chemins de $\overline{C}'$. Comme $\overline{C}'$ respecte les chemins de P' générés par (xv), on sait que $(v_1, w_1) \prec_{\overline{C}'} (w_1, w_2) \prec_{\overline{C}'} (w_2, w_3)$. On en déduit alors que le chemin $((v_1, w_1), (w_1, w_2), (w_2, w_3), (w_3, w_1), (w_1, v_2))$ est un sous-chemin de $\overline{C}'$. De plus, puisque $\overline{C}'$ respecte $((v_1, w_1), (w_1, w_2), (w_2, v_3))$ considéré dans (xv), le chemin $((v_1, w_1), (w_1, w_2), (w_2, w_3), (w_3, w_1), (w_1, v_2))$ apparaît avant le chemin $((v_2, w_2), (w_2, v_3))$ dans $\overline{C}'$. Soit $\overline{C}$ le circuit Eulérien de H obtenu à partir de $\overline{C}'$ en remplaçant respectivement les chemins $((v_1, w_1), (w_1, w_2), (w_2, w_3), (w_3, w_1), (w_1, v_2))$ et $((v_2, w_2), (w_2, v_3))$ par les arcs a_1 et a_2. Comme tous les chemins de P' sauf $((w_1, w_2), (w_2, w_3))$ et $((v_2, w_2), (w_2, v_3))$ sont dérivés de tous les chemins de P, il est clair que $\overline{C}$ respecte P, ce qui termine la preuve. □

Un résultat similaire peut être obtenu pour le nombre total de prédécesseurs $\pi_3(P)$ associés aux arcs de $\mathcal{P}_3(P)$. Le lemme suivant est donné sans preuve.

Lemme 4.9 *Considérons un graphe Eulérien $H = (U, B)$, un sommet u_0 de H et un ensemble P de chemins élémentaires de H. Si $\pi_3(P) \geq 3$, l'instance (H, u_0, P) du PCECPC peut être réduite en temps polynomial en une instance (H', u_0, P') du PCECPC dans laquelle $\sigma_3(P') \leq \sigma_3(P)$ et $\pi_3(P') \leq \pi_3(P) - 1$.* □

Nous donnnons maintenant notre principal résultat de complexité.

Théorème 4.10 *Le problème du circuit Eulérien avec contraintes de précédence induites par des chemins est NP-complet, même si l'ensemble P des chemins élémentaires vérifie les conditions suivantes :*

1) *tout arc a au plus deux successeurs et prédecesseurs dans P, i.e., $|\mathcal{S}_P(a)| + |\mathcal{P}_P(a)| \leq 2$ pour tout $a \in B$,*
2) *la longueur de chaque chemin de P est borné par deux.*

Preuve. Considérons une instance (H, u_0, P) du PCECPC. Par les lemmes 4.8 et 4.9, il existe une réduction polynomiale qui transforme (H, u_0, P) en une autre instance (H', u_0, P') du PCECPC telle que $\mathcal{S}_3(P') = \emptyset$, $\mathcal{P}_3(P') = \emptyset$, et (H', v_0, P') est aussi difficile à résoudre que (H, u_0, P). (Notons que les cardinalités des deux ensembles $\sigma_3(P)$ et $\pi_3(P)$ sont au plus $|A|^2$.)
Pour obtenir une instance satisfaisant les conditions du théorème, on doit transformer (H', u_0, P') en une nouvelle instance $(\overline{H}, u_0, \overline{P})$ du PCECPC comme suit. (Voir la figure

4.9, où $\mathcal{S}_{P'}(a_1) = \{a_2, a_3\}$, et les arcs représentés par des tirets ne peuvent appartenir à $\overline{P}$.) Soit $B'_{P'}$ l'ensemble des arcs de B' qui apparaissent dans P'. Le graphe $\overline{H} = \big(\overline{U}, \overline{B}\big)$ est obtenu à partir de H' en remplaçant séquentiellement chaque arc $a = (u_a, v_a)$ de $B'_{P'}$ par le chemin $((u_a, w_a), (w_a, z_a), (z_a, v_a))$, où w_a et z_a sont deux nouveaux sommets. Chaque chemin $Q = (a_1, a_2, \ldots, a_s)$ de P' est remplacé dans $\overline{P}$ par $(s-1)$ chemins élémentaires $\big((z_{a_i}, v_{a_i}), (v_{a_{i+1}}, w_{a_{i+1}})\big)$, $i = 1, 2, \ldots, s-1$. (Rappelons que $v_{a_i} = u_{a_{i+1}}$ pour tout $i = 1, 2, \ldots, k-1$.)

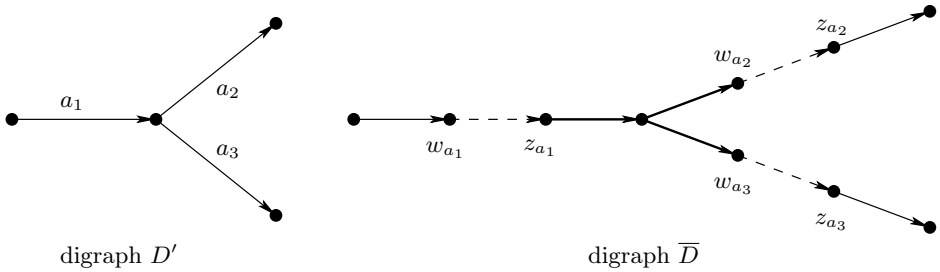

FIGURE 4.9 – Transformation des arcs de $B'_{P'}$; $S_{P'}(a_1) = \{a_2, a_3\}$, les arcs en trait plein peuvent appartenir à $\overline{P}$, les arcs en gras appartiennent à $\overline{P}$, et les arcs en pointillés ne peuvent appartenir à $\overline{P}$

Encore une fois, cette transformation peut être faite en temps polynomial. Comme $\mathcal{S}_3(P') = \emptyset$ et $\mathcal{P}_3(P') = \emptyset$, on a aussi $\mathcal{S}_3(\overline{P}) = \emptyset$ et $\mathcal{P}_3(\overline{P}) = \emptyset$. De plus, d'après la génération des chemins de $\overline{P}$, on sait qu'aucun des arcs (w_a, z_a), $a \in B'_{P'}$, apparaît dans un chemin de $\overline{P}$. Par conséquent, $\overline{P}$ satisfait clairement les deux conditions du théorème.
Comme les sommets de $\{w_a, z_a : a \in B'_{P'}\}$ ont leur degrés entrant et sortant égaux à un dans $\overline{H}$, tout circuit Eulérien de $\overline{H}$ contient $((u_a, w_a), (w_a, z_a), (z_a, v_a))$ comme sous-chemin, pour tout $a \in B'_{P'}$. Il s'ensuit que tout circuit Eulérien de H' respecte P' si et seulement s'il existe un circuit Eulérien de $\overline{H}$ respectant $\overline{P}$. L'instance $(\overline{H}, u_0, \overline{P})$ est donc aussi difficile à résoudre que l'instance (H, u_0, P). □

Soit (H, u_0, P) une instance du PCECPC. Tout sommet v de H peut être l'extrémité de plusieurs chemins de P , disons $Q_1, Q_2, \ldots, Q_s$, $s \geq 2$. (Notons que v peut être le sommet initial d'un chemin de P et le sommet terminal d'un autre chemin.) L'instance (H, u_0, P) peut être transformée en une instance (H', u_0, P') du PCECPC aussi difficle à résoudre dans laquelle chaque sommet est l'extrémité au plus un chemin. En fait, considérons tout chemin ayant v comme extrémité ; sans perte de généralité, considérons Q_1 et supposons que v est le sommet terminal de Q_1. L'arc (u, v) de P_1 incident à v peut être subdivisé en ajoutant un nouveau sommet w. L'ensemble de chemins élémentaires

P peut être transformé en P' en remplçant (u, w) pour (u, v) dans Q_1, et $((u, w), (w, v))$ pour (u, v) dans les chemins $\{Q_i : i = 2, 3, \ldots, s\}$ ayant v comme extrémité. Dans cette nouvelle instance, v est l'extrémité des $s - 1$ chemins de P'. Cette transformation peut augmenter la longueur de certains chemins à respecter. Il peut également augmenter de un le nombre de successeurs et prédécesseurs d'un arc incident à l'arc (u, v) dans les chemins de $P \backslash \{Q_i : i = 1, 2, \ldots, s\}$. De plus, si (H, u_0, P) satisfait les conditions (1)-(2) du théorème 4.10, alors en appliquant séquentiellement la précédente transformation, on obtient une instance du PCECPC dans laquelle chaque sommet est l'extrémité d'au plus un chemin, et le nombre de successeurs et prédécesseurs de chaque arc est au plus trois.

Corollaire 4.11 *Le problème du circuit Eulérien avec contraintes de précédence induites par des chemins est NP-complet, même si l'ensemble de chemins élémentaires P vérifie les conditions suivantes :*

1) *chaque arc a au plus trois prédécesseurs et successeurs dans P, i.e., $|\mathcal{S}_P(a)| + |\mathcal{P}_P(a)| \leq 3$ pour tout $a \in B$,*
2) *chaque sommet est l'extrémité d'au plus un chemin de P.* □

Nous venons de voir que le problème du circuit Eulérien avec contraintes de précédence induites par des chemins est NP-complet. Dans la partie suivante, on montre que le PCECPC peut être résolu en temps polynomial si tous les arcs ont au plus un successeur dans P ou s'ils ont tous au plus un prédécesseur dans P.

4.1.2.2 Cas polynomial

Dans cette section, on considère une instance (H, u_0, P) du PCECPC où $H = (U, B)$ est un graphe Eulérien, u_0 est un sommet de U et P est un ensemble de chemins élémentaires de H. On remarque qu'il existe un cas polynomial trivial du PCECPC lorsque chaque chemin de P est composé d'un arc. En fait, ce problème n'est rien d'autre que le problème du circuit Eulérien dans H. À partir de maintenant, on considère que tous les chemins de P sont de longueur au moins deux.

La condition impliquant que chaque arc de H a au plus un successeur dans P peut être interprété comme la suppression des configurations *Y-sortant* dans P. (Les arcs a_1, a_2 et a_3 de la figure 4.9 forme un Y-sortant puisque $\mathcal{S}_P(a1) = \{a_2, a_3\}$.) On dit que P est *sans-Y-sortant* si P ne contient aucune configuration Y-sortant, *i.e.*, $\max\{|\mathcal{S}_P(a)| : a \in B\} \leq 1$. La figure 4.10 montre un exemple d'un ensemble de chemins élémentaires

sans-Y-sortant. (les trois chemins sont respectivement représentés par des traits plein, des traits pointillés et des tirets.) On peut remarquer que si tous les arcs ont au plus un successeur, l'arc (v_4, v_0) a deux prédécesseurs, à savoir (v_2, v_4) et (v_3, v_4).

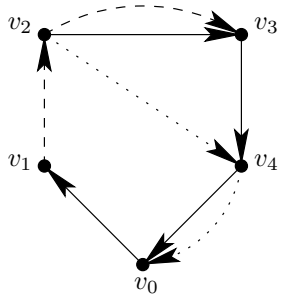

FIGURE 4.10 – Un ensemble de chemins élémentaires sans-Y-sortant

De la même manière, une configuration *Y-entrant* correspond à un arc de P ayant au minimum deux prédécesseurs. P est dit *sans-Y-entrant* si P ne contient pas de configuration Y-entrant, *i.e.*, chaque arc de B a au plus un prédécesseur dans P. Autrement dit, $\max\{|\mathcal{P}_P(a)| : a \in B\} \leq 1$.

Étant donnée une instance (H, u_0, P) du PCECPC, on définit l'*instance inverse* de (H, u_0, P) l'instance définie par le triplet (H^{-1}, u_0, P^{-1}) construit comme suit. Le graphe $H^{-1} = (U, B^{-1})$ est le graphe obtenu en inversant l'orientation des arcs de $H = (U, B)$. Autrement dit, (u, v) est un arc de B^{-1} si et seulement si (v, u) est un arc de B. L'ensemble de chemin P^{-1} est obtenu en parcourant les chemins dans le sens inverse, *i.e.*, $((v_1, v_2), (v_2, v_3), \ldots, (v_{z-1}, v_z)) \in P^{-1}$ si et seulement si $((v_z, v_{z-1}), (v_{z-1}, v_{z-2}), \ldots, (v_2, v_1)) \in P$. Il est clair que (H^{-1}, u_0, P^{-1}) est une instance du PCECPC. En effet, le graphe H^{-1} est Eulérien, u_0 est un sommet de H^{-1} et P^{-1} est un ensemble de chemins élémentaires de H^{-1}. On montre dans la proposition suivante le lien entre instance et instance inverse.

Proposition 4.12 *Soit (H, u_0, P) une instance du PCECPC. Cette instance admet une solution réalisable si et seulement si l'instance inverse (H^{-1}, u_0, P^{-1}) admet elle aussi une solution réalisable.*

Preuve. Soit C une solution réalisable de (H, u_0, P), *i.e.*, un circuit Eulérien commençant en u_0 et respectant P. Considérons le circuit inverse C^{-1} correspondant au circuit obtenu en parcourant C dans le sens inverse. Si $C = ((v_0, v_1), (v_1, v_2), \ldots, (v_z, v_0))$ alors $C^{-1} = ((v_0, v_z), (v_z, v_{z-1}), \ldots, (v_1, v_0))$. C^{-1} est alors un circuit Eulérien commençant

en u_0. Supposons que C^{-1} ne respecte pas P^{-1}. Comme C^{-1} est Eulérien, cela signifie qu'il existe deux arcs d'un chemin Q^{-1} de P^{-1}, disons a_1 et a_2, tels que $a_1 \prec_{Q^{-1}} a_2$ et $a_2 \prec_{C^{-1}} a_1$. Mais dans ce cas, le chemin Q de P associé à Q^{-1} est tel que que $a_2 \prec_Q a_1$ et C vérifie $a_1 \prec_C a_2$, ce qui implique que C n'est pas une solution réalisable de (H, u_0, P). On montre de la même manière qu'étant donnée une solution C de (H^{-1}, u_0, P^{-1}), le circuit inverse est solution de (H, u_0, P), ce qui termine la preuve. □

On remarque que si l'ensemble de chemins P de l'instance (H, u_0, P) est sans-Y-sortant (respectivement sans-Y-entrant), alors l'ensemble P^{-1} de l'instance inverse (H^{-1}, u_0, P^{-1}) est sans-Y-entrant (respectivement sans-Y-sortant).

Soit (H, u_0, P) un triplet tel que $H = (U, B)$ est un graphe orienté quelconque, u_0 est un sommet de U et P est un ensemble de chemins élémentaires sans-Y-sortant. On remarque que si H est Eulérien, alors le triplet (H, u_0, P) correspond à une instance du PCECPC. On introduit maintenant la notion de sous-graphe imprenable qui est au cœur de l'algorithme polynomial. Soit $H' = (U', B')$ un sous-graphe de H. Un sommet v de U est dit H'-*imprenable* par rapport au triplet (H, u_0, P) si pour tout arc a' de $\delta^{\text{in}}_{H'}(v)$, il existe un arc a de $\delta^{\text{out}}_{H'}(v)$ tel que

(i) $a' \prec_P a$ si $v = v_0$,

(ii) soit $a' \prec_P a$ ou $\delta^{\text{out}}_{H \setminus H'}(v) = \emptyset$, si $v \neq v_0$.

Le graphe H' est dit *imprenable* par rapport au triplet (H, u_0, P) si H' contient au minimum un arc mais aucun sommet isolé et si tous ses sommets sont H'-imprenables (par rapport à (H, u_0, P)). On dit alors que le triplet (H, u_0, P) contient un sous-graphe imprenable. Il est possible de tester en temps polynomial si le triplet (H, u_0, P) contient un sous-graphe imprenable. Ceci peut être fait à l'aide de l'algorithme suivant.

Proposition 4.13 *L'algorithme de détection d'un sous-graphe imprenable fonctionne correctement en $O(mnt)$.*

Preuve. Supposons qu'à la fin de l'algorithme, $B' \neq \emptyset$. Alors le graphe retourné est imprenable par rapport à (H, u_0, P). En effet, tous les sommets de $D[B']$ n'apparaissant pas dans la pile à un moment donné de l'algorithme sont des sommets n'ayant aucun arc sortant appartenant à $B \setminus B'$. Ceci implique que pour tout sommet v de cet ensemble, on a $\delta^{\text{out}}_{H \setminus D[B']}(v) = \emptyset$. Ces sommets sont donc $D[B']$-imprenables par rapport à (H, u_0, P). Soit v un sommet de $D[B']$ tel que v est apparu dans la pile durant l'algorithme. La dernière fois que v est considéré dans la boucle tant que, on supprime de B' tous les

Algorithme 5: Algorithme de détection d'un sous-graphe imprenable

Entrée : - Graphe orienté $H = (U, B)$ avec $|B| = m$,
- sommet u_0 de U,
- ensemble P sans-Y-sortant de t chemins élémentaires de H.

Sortie : Soit un sous-graphe imprenable par rapport à (H, u_0, P), soit un certificat assurant que (H, u_0, P) ne contient pas de sous-graphe imprenable.

Début

```
B' = B;
Pile S = {u_0};
tant que S ≠ ∅ faire
    v = Dépiler(S);
    pour ∀ (u, v) ∈ δ^in_{B'}(v) faire
        si ∄ (v, w) ∈ B' avec (u, v) ≺_P (v, w) alors
            B' = B' \ {(u, v)};
            Empiler(S, u);
si B' ≠ ∅ alors
    retourner D[B'];
sinon
    retourner Certificat;
```

Fin

arcs a_1 entrant dans v pour lesquels il n'existe pas dans B' un arc a_2 sortant de v avec $a_1 \prec_P a_2$. Donc, à la fin de l'algorithme, tous les arcs entrants de v dans B' ont un successeur dans $B' \cap \delta^{\text{out}}(v)$, ce qui implique que le sommet v est $D[B']$-imprenable. Par ailleurs, par définition de l'algorithme, si à la fin, B' est vide, cela signifie qu'il n'existe pas de sous-graphe imprenable par rapport à (H, u_0, P). En effet, l'algorithme ne peut pas supprimer un arc d'un sous-graphe imprenable. L'algorithme fonctionne donc correctement.

Pour terminer la preuve, on doit maintenant déterminer la complexité de l'algorithme. Comme chaque fois que l'on empile un sommet dans S, cela signifie que l'on a supprimé un arc de B, le nombre d'itération de la boucle while est au maximum m. De plus, comme savoir si un arc (u, v) possède un successeur dans P parmi les arcs de $\delta^{\text{out}}_{B'}(v)$ peut être fait en $o(t)$, et comme chaque sommet possède au plus n arcs entrants, il est clair que l'algorithme fonctionne en $O(mnt)$. La complexité de cet algorithme provient du fait qu'il peut être nécessaire de tester plusieurs fois si un arc posède un successeur dans K puisque celui-ci peut être supprimé à tout moment durant l'algorithme. □

Le reste de cette section s'attache à montrer que le PCECPC peut être résolu en temps polynomial si l'ensemble de chemins élémentaires P de l'instance (H, u_0, P) est sans-Y-sortant. Soit (H, u_0, P) un triplet où $H = (U, B)$ est un graphe orienté quelconque, u_0 un sommet particulier de U et P un ensemble de chemins élémentaires de H sans-Y-Sortant. Soit a un arc de B. On note par $(H \setminus a, u_0, P \setminus a)$ le triplet obtenu à partir de (H, u_0, P) en supprimant dans H l'arc a (si après suppression, il existe un sommet isolé, alors ce dernier est également supprimé) et en supprimant dans chaque chemin de P l'arc a. (La suppression de l'arc a dans un chemin Q de P peut générer deux chemins distincts si $a \in Q$ mais que a n'est pas le premier ni le dernier arc dans Q). On remarque cependant que si P est sans-Y-sortant, alors $P \setminus a$ l'est aussi. On donne maintenant quelques propriétés reliant les triplets (H, u_0, P) et $(H \setminus a, u_0, P \setminus a)$.

Proposition 4.14 *Soit (H, u_0, P) un triplet où $H = (U, B)$ est un graphe orienté quelconque, u_0 un sommet particulier de U et P un ensemble de chemins élémentaires de H sans-Y-sortant. Soit a un arc de $\delta^{\text{out}}(u_0)$. Si (H, u_0, P) ne contient pas de sous-graphe imprenable, alors $(H \setminus a, u_0, P \setminus a)$ non plus.*

Preuve. Supposons que $(H \setminus a, u_0, P \setminus a)$ contient un sous-graphe imprenable, disons $H' = (U', B')$. Notons par U_1 l'ensemble des sommets de U' tels que $\delta^{\text{out}}_{(H\setminus a)\setminus H'}(v) \neq \emptyset$. On ajoute également le sommet u_0 dans U_1 si $u_0 \in U'$. Comme H' est imprenable, tous les sommets v de U_1 sont tels que pour tout arc a' de $\delta^{\text{in}}_{H'}(v)$, il existe un arc sortant a de $\delta^{\text{out}}_{H'}(v)$ avec $a' \prec_P a$. Il est clair que ces sommets sont aussi H'-imprenables par rapport à (H, u_0, P). Par ailleurs, comme u_0 n'appartient pas à $U' \setminus U_1$, il est clair que l'on a $\delta^{\text{out}}_{H\setminus H'}(v) = \emptyset$ pour tout sommet v de $U' \setminus U_1$, ce qui implique que ces derniers sont H'-imprenables par rapport à (H, u_0, P). Par conséquent, H' est un sous-graphe imprenable de (H, u_0, P), ce qui contredit l'hypothèse de départ. □

Proposition 4.15 *Soit (H, u_0, P) un triplet où $H = (U, B)$ est un graphe orienté quelconque, u_0 un sommet particulier de U et P un ensemble de chemins élémentaires de H sans-Y-sortant. Soit v_1 un sommet de $U \setminus \{u_0\}$ tel que $\deg^{\text{in}}(v_1) < \deg^{\text{out}}(v_1)$. Soit $\bar{B}$ l'ensemble des arcs sortants de v_1 n'ayant pas de prédédesseur dans P. Si (H, u_0, P) ne contient pas de sous-graphe imprenable, alors il existe un arc a appartenant à $\bar{B}$ tel que $(H \setminus a, u_0, P \setminus a)$ ne contient pas de sous-graphe imprenable.*

Preuve. Tout d'abord, il est facile de voir que l'ensemble $\bar{B}$ est différent de l'ensemble vide. Autrement, cela signifie que tout arc a sortant de v_1 a un prédécesseur dans P. Comme P est un ensemble de chemins élémentaires, ceci implique qu'il existe un arc

a' entrant dans v_1 tel que $a' \prec_P a$. Comme P est sans-Y-sortant, chaque arc possède au maximum un successeur dans P. Finalement, étant donné que l'on a $\deg^{\text{in}}(v_1) < \deg^{\text{out}}(v_1)$, $\bar{B}$ contient au moins un arc, disons a_1.

Supposons que $(H \setminus a_1, u_0, P \setminus a_1)$ contient un sous-graphe imprenable, disons $H' = (U', B')$. Notons par U_1 l'ensemble des sommets de U' tels que tout arc entrant de v dans B' possède un arc sortant de v dans B' comme successeur dans P. Par définition, tous les sommets de U_1 sont H'-imprenables par rapport à (H, u_0, P) puisque l'arc a_1 n'appartient pas à H'. Considérons maintenant un sommet v de $U' \setminus U_1$ différent de v_1. Comme v est H'-imprenable par rapport à $(H \setminus a_1, u_0, P \setminus a_1)$, il s'ensuit que $\delta^{\text{out}}_{(H \setminus a_1) \setminus H'}(v) = \emptyset$. Comme a_1 n'est pas un arc sortant de v, alors on a $\delta^{\text{out}}_{H \setminus H'}(v) = \emptyset$ et v est alors H'-imprenable par rapport à (H, u_0, P). Comme H' n'est pas un sous-graphe imprenable par rapport à (H, u_0, P), on déduit que v_1 appartient à $U' \setminus U_1$. On obtient que tout arc sortant de v_1 différent de l'arc a_1 appartient à H' puisque v_1 est H'-imprenable par rapport à $(H \setminus a_1, u_0, P \setminus a_1)$ et $v_1 \notin U_1$. Par ailleurs, comme $\deg^{\text{in}}(v_1) < \deg^{\text{out}}(v_1)$, P est sans-Y-sortant, et il existe un arc a' de $\delta^{\text{in}}_{H'}(v_1)$ sans successeur dans P parmi les arcs de $\delta^{\text{out}}(v_1)$, alors il existe un arc de $\delta^{\text{out}}(v_1) \setminus \{a_1\}$, disons a_2, appartenant à $\bar{B}$. De plus, a_2 est un arc de H'.

Supposons maintenant que $(H \setminus a_2, u_0, P \setminus a_2)$ contient un sous-graphe imprenable, disons $H'' = (U'', B'')$. En appliquant le même raisonnement, on déduit que a_1 est un arc de H''. Notons par $H^* = (U^*, B^*)$ le graphe induit par la réunion de H' et H'', *i.e.*, $H^* = H' \cup H''$. On montre que H^* est un sous-graphe imprenable de (H, u_0, P). Supposons le contraire. Il existe alors $v \in U^*$ tel que v n'est pas H^*-imprenable par rapport à (H, u_0, P). Il est clair que v est différent de v_1. En effet, on a montré que tout arc de $\delta^{\text{out}}(v_1) \setminus \{a_1\}$ (respectivement $\delta^{\text{out}}(v_1) \setminus \{a_2\}$) appartient à H' (respectivement H''), ce qui implique $\delta^{\text{out}}_{H \setminus H^*}(v_1) = \emptyset$. Comme v_1 est différent de u_0, alors v_1 est H^*-imprenable par rapport à (H, u_0, P). Supposons maintenant que v est un sommet de $U' \setminus \{v_1\}$. Si v n'est pas H^*-imprenable par rapport à (H, u_0, P), cela implique qu'il existe dans B^* un arc entrant de v, disons $\bar{a}$, n'ayant pas un arc sortant de v dans B^* comme successeur dans P. De plus, si $v \neq u_0$, alors on a $\delta^{\text{out}}_{H \setminus H^*}(v) \neq \emptyset$. Supposons que $\bar{a}$ est un arc de H'. Il n'existe alors pas dans H' un arc sortant a de v avec $\bar{a} \prec_P a$. Ceci implique qu'il n'existe pas dans H' un arc sortant a de v avec $\bar{a} \prec_{P \setminus \{a_1\}} a$. De plus, si $v \neq u_0$, il est facile de voir que $\delta^{\text{out}}_{(H \setminus a_1) \setminus H'}(v) \neq \emptyset$ puisque a_1 n'est pas un arc sortant de v. Ceci contredit alors l'hypothèse que H' est un sous-graphe imprenable de $(H \setminus a_1, u_0, P \setminus a_1)$. Si $\bar{a}$ est un arc de H'', on déduit de manière similaire que H'' n'est pas un sous-graphe imprenable de $(H \setminus a_2, u_0, P \setminus a_2)$, ce qui termine la preuve. □

Proposition 4.16 *L'algorithme PCECPC fonctionne correctement en* $O(m^2n^2t)$.

Algorithme 6: Algorithme PCECPC

Entrée : - Graphe orienté Eulérien $H = (U, B)$ avec $|U| = n$, $|B| = m$,
- sommet u_0 de U,
- ensemble P sans-Y-sortant de t chemins élémentaires de H.

Sortie : Soit un sous-graphe imprenable, soit une solution réalisable pour l'instance (H, u_0, P).

Début

  si (H, u_0, P) *contient un sous-graphe imprenable* H' **alors**
    retourner H';
  $C = \emptyset$;
  $h = u_0$;
  tant que $B \neq \emptyset$ **faire**
    Choisir (h, v) tel que $(H \setminus (h, v), u_0, P \setminus (h, v))$ ne contient pas de sous-graphe imprenable et $(C, (h, v))$ respecte P;
    $C = (C, (h, v))$;
    $P = P \setminus (h, v)$;
    $H = H \setminus \{(h, v)\}$;
    $h = v$;
  retourner C;

Fin

Preuve. Supposons que (H, u_0, P) contient un sous-graphe imprenable, disons $H' = (U', B')$. Considérons l'instance inverse (H^{-1}, u_0, P^{-1}) de (H, u_0, P). Le sous-graphe H'^{-1} associé à H' est tel que pour tout sommet $v \in H'^{-1}$, et pour tout arc a de $\delta^{\text{out}}_{H'}(v)$, il existe un arc entrant a' de $\delta^{\text{in}}_{H'}(v)$ tel que

(i) $a' \prec_P a$ si $v = u_0$,

(ii) soit $a' \prec_P a$ ou $\delta^{\text{in}}_{H \setminus H'}(v) = \emptyset$, si $v \neq u_0$.

Considérons un arc (u, v) de H'^{-1}. Étant donné un chemin $C = ((u_1, u_2), (u_2, u_3), \ldots, (u_z, u))$ respectant P^{-1}, $(C, (u, v))$ est un chemin respectant P^{-1} si et seulement si l'arc (u_z, u) appartient lui aussi à H'^{-1}. Ceci implique que le circuit commençant en u_0 ne peut traverser aucun arc de H'^{-1} tout en respectant P^{-1}. L'instance (H^{-1}, u_0, P^{-1}) n'admet donc pas de solution. Par conséquent, (H, u_0, P) n'admet pas non plus de solution.

Supposons maintenant que (H, u_0, P) ne contient pas de sous-graphe imprenable. On montre que l'algorithme PCECPC retourne à la fin un circuit Eulérien C commençant en u_0 et respectant P. Pour cela, il suffit de montrer qu'à chaque itération, il existe (h, v) tel que $(H \setminus (h, v), u_0, P \setminus (h, v))$ ne contient pas de sous-graphe imprenable et $(C, (h, v))$ respecte P. Comme C commence en u_0 et qu'à la fin de l'algorithme, tous les

arcs de B ont été ajoutés exactement une fois dans C, on obtient que C est une solution réalisable de (H, u_0, P). Soit C le chemin obtenu après un certain nombre d'itérations. h représente l'extrémité terminale de C (si C est vide, alors $h = u_0$). Notons par $\bar{B}$ l'ensemble des arcs sortants de h tels que pour tout arc $a \in \bar{B}$, il n'existe pas $a' \in \delta^{\text{in}}(h)$ avec $a' \prec_P a$. Comme à chaque itération, les arcs du chemin C sont supprimés dans le graphe H, alors pour tout arc $a \in \bar{B}$, (C, a) respecte P. De plus, il est clair qu'à chaque itération, P est sans-Y-sortant et (H, u_0, P) ne contient pas de sous-graphe imprenable.

Supposons que $h = u_0$. Il est clair que $\bar{B}$ est différent de l'ensemble vide. Autrement, comme $\deg^{\text{in}}(u_0) = \deg^{\text{out}}(u_0)$ et P est sans-Y-sortant, cela signifie que pour tout arc entrant a' de u_0, il existe $a \in \delta^{\text{out}}(u_0)$ avec $a' \prec_P a$. Le sommet u_0 est alors H-imprenable par rapport à (H, u_0, P). Ceci implique que H est un sous-graphe imprenable par rapport à (H, u_0, P), ce qui contredit l'hypothèse que (H, u_0, P) ne contient pas de sous-graphe imprenable. Comme $\bar{B} \neq \emptyset$, la proposition 4.14 implique que pour tout arc $a \in \bar{B}$, $(H \setminus a, u_0, P \setminus a)$.

Supposons maintenant que h est différent de u_0. Comme le graphe courant H est obtenu en supprimant du graphe initial les arcs du chemin C commençant en u_0, on a alors $deg^{\text{in}}(h) = deg^{\text{out}}(h) - 1$. La proposition 4.15 assure qu'il existe un arc (h, v) tel que $(H \setminus (h, v), u_0, P \setminus (h, v))$ ne contient pas de sous-graphe imprenable et $(C, (h, v))$ respecte P.

Pour terminer la preuve, il suffit de prouver la complexité de cet algorithme. La boucle principale s'exécute au maximum m fois. À chaque itération, on teste pour tout arc sortant a de h si le triplet $(H \setminus a, u_0, P \setminus a)$ ne contient pas de sous-graphe imprenable et si (C, a) respecte P. Comme la complexité de ces opérations est respectivement de $O(mnt)$ et $O(t)$, et que chaque arc possède au maximum n arcs entrants, la complexité de chaque itération est en $O(mn^2t)$. □

L'algorithme du PCECPC permet de donner les conditions nécessaires et suffisantes pour qu'une instance du PCECPC admette une solution réalisable si l'ensemble de chemins élémentaires est sans-Y-sortant. Ces conditions sont énoncées dans le théorème suivant.

Théorème 4.17 *Soit (H, u_0, P) une instance du PCECPC. Si P est sans-Y-sortant, alors (H, u_0, P) admet une solution réalisable si et seulement si (H, u_0, P) ne contient pas de sous-graphe imprenable.* □

Comme toute instance (H, u_0, P) admet une solution si et seulement si l'instance inverse (H^{-1}, u_0, P^{-1}) en admet une, le PCECPC peut également être résolu en temps

polynomial par notre algorithme si l'ensemble de chemins est sans-Y-entrant. Il convient cependant de modifier la définition d'un graphe imprenable. Soit (H, u_0, P) une instance du PCECPC avec P sans-Y-entrant. Soit $H' = (U', B')$ un sous-graphe de H. Un sommet v de U est dit H'-*imprenable* par rapport au triplet (H, u_0, P) si pour tout arc a de $\delta^{\text{out}}_{H'}(v)$, il existe un arc entrant a' de $\delta^{\text{in}}_{H'}(v)$ tel que

(i) $a' \prec_P a$ si $v = u_0$,

(ii) soit $a' \prec_P a$ ou $\delta^{\text{in}}_{H \setminus H'}(v) = \emptyset$, si $v \neq u_0$.

Le graphe H' est dit *imprenable* par rapport au triplet (H, u_0, P) si H' contient au minimum un arc mais aucun sommet isolé et si tous ses sommets sont H'-imprenables (par rapport à (H, u_0, P)). On obtient alors le corollaire suivant.

Corollaire 4.18 *Soit (H, u_0, P) une instance du ECWPPCP avec P sans-Y-entrant. Cette instance admet une solution si et seulement si (H, u_0, P) ne contient pas de sous-graphe imprenable. De plus, le problème peut être résolu en temps polynomial.*

Preuve. Considérons l'instance inverse (H^{-1}, u_0, P^{-1}) de (H, u_0, P). Il est clair que P^{-1} est sans-Y-sortant. On peut donc résoudre le PCECPC pour l'instance (H^{-1}, u_0, P^{-1}) en temps polynomial. Supposons (H^{-1}, u_0, P^{-1}) admette une solution réalisable, disons C^{-1}. En parcourant le circuit dans le sens inverse, on obtient une solution réalisable pour l'instance (H, u_0, P). Par ailleurs, supposons que (H^{-1}, u_0, P^{-1}) n'admet pas de solution réalisable. Cela signifie que (H^{-1}, u_0, P^{-1}) contient un sous-graphe imprenable, disons H'^{-1}. Le sous-graphe H' obtenu en inversant l'orientation des arcs de H^{-1} est alors imprenable par rapport à (H, u_0, P). Comme la construction de (H^{-1}, u_0, P^{-1}) en fonction de (H, u_0, P) se fait en temps polynomial et que chaque instance admet une solution si et seulement l'autre instance en admet également une, le PCECPC peut être résolu en temps polynomial si l'ensemble de chemins élémentaires de l'instance est sans-Y-entrant. De plus, l'instance (H, u_0, P) avec P sans-Y-sortant admet une solution si et seulement si elle ne contient pas de sous-graphe imprenable. □

On vient de montrer que dans le cas général, le problème du circuit Eulérien avec contraintes de précédence induites par des chemins est un problème NP-complet. Cependant, si chaque arc a au plus un successeur ou un prédécesseur dans L, le problème peut être résolu en temps polynomial. Comme ce problème est équivalent au problème de satisfiabilité de la séquence du véhicule, on a prouvé que, dans le cas général, la représentation des solutions du 1-PRLPA décrivant les solutions à l'aide des ensembles des arcs du véhicule et des demandes n'est pas une représentation manipulable puisqu'elle ne permet de vérifier en temps polynomial si une solution du 1-PRLPA est réalisable.

Cependant, supposons que le véhicule transporte une seule demande à la fois. Dans ce cas, les chemins des demandes sont arc-disjoints. Chaque arc possède au maximum un successeur et un prédécesseur et le problème de satisfiabilité peut alors être résolu en temps polynomial. Ceci implique que la représentation est alors manipulable.

Ces différents résultats de complexité impliquent que les solutions du 1-PRLPA doivent être décrites par l'ensemble des arcs du véhicule, l'ordre dans lequel ces derniers sont traversés et les arcs traversés par chaque demande. De plus, cette représentation est minimale puisque l'on ne peut supprimer une de ces informations sans perdre la polynomialité du problème de satisfiabilité d'une solution. Ceci signifie que toute formulation du 1-PRLPA sous forme de programme linéaire en nombres entiers doit contenir des variables permettant de représenter ces trois types information.

Considérons le problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire, lorsque le véhicule ne transporte qu'une demande à la fois. Dans ce problème, comme les chemins des demandes sont arc-disjoints, on peut tester en temps polynomial si une solution est réalisable lorsqu'elle est décrite uniquement par les ensembles d'arcs du véhicule et des demandes. De plus, cette représentation est minimale. Une formulation du 1-PRLPA unitaire peut alors représenter toute solution en codant uniquement les ensembles d'arcs et ne plus avoir de variables correspondant à l'ordre dans lequel les arcs du véhicule sont traversés. Par contre, comme le prouve la complexité du problème de satisfiabilité des demandes, on doit avoir des variables codant les arcs traversés par les demandes, que ce soit pour le cas général ou pour le cas unitaire.

Nous terminons ce chapitre en donnant deux formulations sous forme de programmes linéaires en nombres entiers modélisant le 1-PRLPA dans le cas unitaire et dans le cas général. Nous utilisons dans les deux cas des variables codant les représentations minimales des solutions du problème. Nous commençons par la formulation associée au cas unitaire puis l'étendons dans le cas général en ajoutant de nouvelles variables et de nouvelles contraintes codant l'ordre des arcs du véhicule.

4.2 Formulation du 1-PRLPA unitaire

Nous présentons dans cette section une formulation du problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire sous forme de programme linéaire en nombres entiers. On rappelle que dans ce problème, le véhicule ne peut transporter qu'une demande à la fois. Comme aucune demande ne peut être transportée avec fractionnement, le 1-PRLPA unitaire survient lorsque $q^k + q^{k'} > Q$ pour toute

paire de demandes distinctes k et k' de K. On fixe, sans perte de généralité, que $Q = 1$ et $q^k = 1$ pour tout $k \in K$.

Avant de continuer, il est nécessaire d'introduire de nouvelles notations. On considère le graphe orienté $\Phi = (V, \{(o^k, d^k) : k \in K\})$, appelé *graphe des demandes*. L'ensemble des sommets de Φ est le même que celui du graphe D et chaque arc de Φ relie l'origine d'une demande de K à sa destination. On peut alors utiliser les notations relatives aux graphes en précisant, en indice, le graphe Φ. Ainsi, étant donné un sous-ensemble $W \subseteq V$, l'ensemble $A_\Phi(W)$ représente l'ensemble des arcs de Φ dont les deux extrémités sont des sommets de W, c'est-à-dire, $A_\Phi(W) = \{(o^k, d^k), k \in K : o^k, d^k \in W\}$. De la même manière, l'ensemble $\delta_\Phi(W)$ représente l'ensemble des arcs associés aux demandes de K ayant exactement une extrémité dans W, *i.e.*, $\delta_\Phi(W) = \{(o^k, d^k), k \in K : |\{o^k, d^k\} \cap W| = 1\}$. Par ailleurs, étant donnés un sous-ensemble d'arcs B de Φ et une demande $k \in K$, on dit que k appartient à B, et l'on note $k \in B$, si l'arc (o^k, d^k) associé à la demande k est un arc de B.

On suppose, sans perte de généralité, que les demandes sont toujours transportées sur des chemins élémentaires. De ce fait, il est clair qu'une demande $k \in K$ ne peut être transportée sur les arcs entrants de o^k ni sur les arcs sortants de d^k, sans créer de circuit. Pour une demande $k \in K$, on note par A^k l'ensemble des arcs pouvant être traversés par la demande k. A^k est alors équivalent à $A \setminus \{\delta^{\text{in}}(o^k) \cup \delta^{\text{out}}(d^k)\}$. Pour tout arc (i, j) de A, on note par $K^{(i,j)}$ l'ensemble des demandes pouvant être transportées sur l'arc (i, j). $K^{(i,j)}$ est alors équivalent à $K^{(i,j)} = \{k \in K : o^k \neq j \text{ et } d^k \neq i\}$. On note par n la cardinalité de V et par m la cardinalité de A. Pour une demande $k \in K$, on note par m_k la cardinalité de A^k et l'on pose $m_K = \sum_{k \in K} m_k$.

Suivant les résultats obtenus dans la partie précédente, les solutions du 1-PRLPA unitaire peuvent être décrites en utilisant uniquement les ensembles d'arcs des chemins des demandes et du circuit du véhicule. On introduit maintenant deux ensembles de variables, le premier représentant les arcs des chemins des demandes et le second représentant les arcs traversés par le véhicule. Soit $x \in \mathbb{R}^{m_K}$ tel que

$$x_a^k = \begin{cases} 1 & \text{si la demande } k \text{ est transportée sur l'arc } a, \\ 0 & \text{sinon,} \end{cases}$$

pour toute demande $k \in K$ et pour tout arc $a \in A^k$, et soit $y \in \mathbb{R}^m$ tel que

$$y_a = \begin{cases} 1 & \text{si le véhicule traverse l'arc } a, \\ 0 & \text{sinon,} \end{cases}$$

pour tout arc $a \in A$. Soit $S_U(D, v_0, K)$ l'ensemble des vecteurs (x, y) associés aux solutions réalisables du 1-PRLPA unitaire. Un vecteur (x, y) de $S_U(D, v_0, K)$ satisfait

les contraintes suivantes

$$\sum_{a\in\delta^{\text{out}}(W)} y_a - y_{a'} \geq 0 \qquad \forall\ W \subseteq V \text{ tel que } v_0 \in W, \forall\ a' \in A(\overline{W}), \quad (4.10)$$

$$\sum_{a\in\delta^{\text{out}}(v)} y_a - \sum_{a\in\delta^{\text{in}}(v)} y_a = 0 \qquad \forall\ v \in V, \quad (4.11)$$

$$\sum_{a\in\delta^{\text{out}}(v)\cap A^k} x_a^k - \sum_{a\in\delta^{\text{in}}(v)\cap A^k} x_a^k = b_v^k \qquad \forall\ k \in K,\ \ \forall\ v \in V, \quad (4.12)$$

$$\sum_{a\in\delta^{\text{out}}(v)\cap A^k} x_a^k \leq 1 \qquad \forall\ k \in K,\ \ \forall\ v \in V \setminus \{o^k, d^k\}, \quad (4.13)$$

$$\sum_{a\in\delta^{\text{out}}(W)\cap A^k} x_a^k - \sum_{a\in\delta^{\text{out}}(v)\cap A^k} x_a^k \geq 0 \qquad \begin{array}{l}\forall k \in K, \forall\ W \subseteq V \text{ tel que } o^k, d^k \in W,\\ \forall\ v \in \overline{W},\end{array} \quad (4.14)$$

$$y_a - \sum_{k\in K^a} x_a^k \geq 0 \qquad \forall\ a \in A, \quad (4.15)$$

$$y_a \geq 0 \qquad \forall\ a \in A, \quad (4.16)$$

$$y_a \leq 1 \qquad \forall\ a \in A, \quad (4.17)$$

$$x_a^k \geq 0 \qquad \forall\ k \in K, \forall\ a \in A^k, \quad (4.18)$$

$$x_a^k \leq 1 \qquad \forall\ k \in K, \forall\ a \in A^k, \quad (4.19)$$

où le nombre b_v^k, défini par

$$b_v^k = \begin{cases} 1 & \text{si } v = o^k, \\ -1 & \text{si } v = d^k, \\ 0 & \text{sinon,} \end{cases}$$

représente le volume de la demande $k \in K$ disponible/requis associé à chaque sommet $v \in V$. Les contraintes (4.10) et (4.11) sont respectivement appelées *contraintes de connexité* et *contraintes de conservation de flot associées au véhicule*. Elles impliquent que l'ensemble d'arcs $A_y = \{a \in A : y_a = 1\}$, correspondant au circuit du véhicule, induit un circuit Eulérien passant par v_0. Les contraintes (4.10) assurent que v_0 est incident à au moins un arc de A_y et que le graphe orienté induit est faiblement connexe. Les contraintes (4.11) sont les contraintes de conservation de flot et forcent le nombre d'arcs entrants de chaque sommet à être égal au nombre d'arcs sortants de ce sommet. Les contraintes (4.12) sont les *contraintes de conservation de flot associées aux demandes*. Elles impliquent que les demandes sont transportées de leur origine à leur destination à l'aide d'un chemin. Les contraintes (4.13), appelées *contraintes de circuit*, empêchent les demandes de passer plus d'une fois par sommet et garantissent alors que

le chemin de chaque demande est élémentaire. Les *contraintes de connexité des demandes* (4.14) impliquent que le graphe induit par les arcs sur lesquels est transportée toute demande de K est connexe. (La validité de ces contraintes provient du fait que les variables x vérifient également les contraintes (4.13).) Les contraintes (4.15) sont les contraintes de capacité associées au véhicule. Elles imposent que le véhicule transporte au maximum une demande par arc. Les contraintes (4.16)-(4.19) sont les contraintes triviales associées aux variables x et y.

On remarque que selon la structure du graphe D, les contraintes de circuit peuvent être renforcées. Ainsi, si le graphe D contient l'arc (o^k, d^k) pour une demande k de K, la contrainte de circuit (4.13) associée à k peut être remplacée par la contrainte

$$x^k_{(o^k,d^k)} + \sum_{a \in \delta^{\text{out}}(v) \cap A^k} x^k_a \leq 1 \qquad \forall\ k \in K,\ \ \forall\ v \in V \setminus \{o^k, d^k\}. \tag{4.20}$$

Soit S l'ensemble des vecteurs binaires (x, y) satisfaisant (4.10)-(4.19). La proposition suivante montre que les contraintes (4.10)-(4.19) ne suffisent pas à définir la formulation du 1-PRLPA unitaire.

Proposition 4.19 $S_U(D, v_0, K) \subsetneq S$

Preuve. L'inclusion vient directement de la validité des contraintes (4.10)-(4.19) pour le 1-PRLPA unitaire. Un vecteur (x, y) de S fournit l'ensemble des arcs du circuit Eulérien (i.e., l'ensemble A_y) représentant le trajet du véhicule, ainsi que les ensembles d'arcs des chemins des demandes. Pour s'assurer que (x, y) est associé à une solution réalisable du 1-PRLPA unitaire, on doit maintenant considérer le problème de satisfiabilité de la séquence du véhicule où l'entrée correspond aux ensembles d'arcs donnés par (x, y). Les conditions nécessaires et suffisantes du théorème 4.17 pour que le PSSV admette une solution réalisable impliquent que le graphe Eulérien induit par A_y ne contient pas de sous-graphe Eulérien imprenable par rapport aux chemins des demandes induits par les variables x. Cette condition n'est pas garantie par les contraintes (4.10)-(4.19), puisque ces dernières n'empêchent pas d'avoir par exemple le cas suivant : le dépôt v_0 a un seul arc entrant a_1 et un seul arc sortant a_2, et il existe un chemin d'une demande qui est transportée sur le chemin (a_1, a_2). Le vecteur (x, y) peut donc induire quelques sous-graphes Eulériens imprenables et donc correspondre à des solutions irréalisables pour le 1-PRLPA unitaire. □

Pour empêcher qu'un vecteur (x, y) de S induise des solutions contenant des sous-graphes Eulériens imprenables, on introduit les inégalités linéaires suivantes.

Proposition 4.20 *Soit $W \neq \emptyset$ un sous-ensemble propre de sommets de V tel que $v_0 \in W$, $A_\Phi(\overline{W}) \neq \emptyset$ et $\delta_\Phi(W) = \emptyset$. La contrainte de vulnérabilité associée à W*

$$\sum_{a \in \delta^{\text{out}}(W)} y_a - \sum_{k \in A_\Phi(\overline{W})} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x_a^k + M \sum_{k \in A_\Phi(W)} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x_a^k \geq 1, \tag{4.21}$$

où M est une constante suffisamment grande, est valide pour le 1-PRLPA.

Preuve. Supposons qu'il existe une solution (x, y) réalisable pour le 1-PRLPA unitaire qui viole une inégalité de type (4.21), c'est-à-dire,

$$\sum_{a \in \delta^{\text{out}}(W)} y_a - \sum_{k \in A_\Phi(\overline{W})} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x_a^k + M \sum_{k \in A_\Phi(W)} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x_a^k \leq 0.$$

Comme (x, y) est réalisable, on sait que $y_a - x_a(A_\Phi(\overline{W}) \cap K^a) \geq 0$ pour tout $a \in A$. On déduit que $x_a(A_\Phi(W) \cap K^a) = 0$ et $y_a = x_a(A_\Phi(\overline{W}) \cap K^a)$ pour tout $a \in \delta^{\text{out}}(W)$. Par conséquent, sur n'importe quel arc a sortant de W avec $y_a = 1$, le véhicule transporte une demande ayant son origine et sa destination dans $\overline{W}$. Comme le dépôt v_0 appartient à W et $A_\Phi(\overline{W}) \neq \emptyset$, cela implique qu'il n'existe pas d'arc sortant de W pour atteindre d'abord l'origine d'une demande de $A_\Phi(\overline{W})$. Le vecteur (x, y) ne peut donc correspondre à un vecteur de $S_U(D, v_0, K)$. □

On prouve maintenant que les contraintes de vulnérabilité (4.21) sont suffisantes pour réduire S à $S_U(D, v_0, K)$.

Proposition 4.21 *Tout vecteur binaire de S satisfaisant les contraintes* (4.21) *induit une solution réalisable pour le 1-PRLPA unitaire.*

Preuve. Soient (x, y) un vecteur de S, D_y le graphe orienté Eulérien induit par l'ensemble d'arcs A_y du trajet du véhicule, et $L = \{L_1, \ldots, L_p\}$ l'ensemble des chemins des demandes induit par les variables x. Comme nous considérons le cas unitaire, les chemins de L sont deux-à-deux arc-disjoints. Rappelons que (x, y) induit une solution réalisable pour le 1-PRLPA unitaire si et seulement si le graphe Eulérien D_y ne contient pas un sous-graphe Eulérien imprenable par rapport à L. On prouve maintenant que l'existence d'un sous-graphe Eulérien imprenable, disons $D' = (V', A')$, de D_y par rapport à L implique que (x, y) viole une contrainte de vulnérabilité (4.21).

Considérons les sous-ensembles de sommets $V_1 = \{v \in V' : v = v_0 \text{ ou } \delta_{D_y}(v) \setminus A' \neq \emptyset\}$ et $V_2 = V' \setminus V_1$. Soit $W = V \setminus V_2$. On remarque que $v_0 \in W$ et V_2 est composé de tous

les sommets de $V' \setminus \{v_0\}$ qui ne sont pas incidents à des arcs de $A_y \setminus A'$. On a alors $V_1 \subseteq W$ et

$$\delta^{\text{out}}_{D_y}(W) \subseteq \bigcup_{v \in V_1} \delta^{\text{out}}_{D'}(v). \tag{4.22}$$

On montre maintenant que toute demande transportée sur un arc de A' a son origine et sa destination dans V_2. Considérons une demande $k \in K$ transportée sur le chemin $L_k = (a_1, a_2, \ldots, a_t)$ avec $t \geq 1$, tel qu'au moins un des arcs de L_k appartient à A'. Supposons que $o^k \notin V_2$ et soit a_j, $j \in \{1, 2, \ldots, t\}$, le premier arc de L_k dans A'. Si $j = 1$, il est clair que $o^k \in V_1$. Comme les chemins des demandes sont deux-à-deux arc-disjoints, l'arc a_1 n'a pas de prédécesseur ce qui signifie que o^k n'est pas D'-imprenable. Si $j \in \{2, 3, \ldots, t\}$, alors l'arc $a_{j-1} = (u, v)$ est dans $A_y \setminus A'$ et $v \in V_1$. L'arc a_j n'a alors pas de prédécesseur dans A' ce qui implique que v n'est pas D'-imprenable. De manière similaire, on prouve que $d^k \in V_2$.

D'après la définition des sous-graphes Eulériens imprenables, on sait que D' contient au moins un arc traversé par une demande de K. L'assertion précédente implique qu'il existe au moins une demande dont l'origine et la destination sont dans V_2. De plus, comme les sommets de V_2 sont uniquement incidents à A', il ne peut exister une demande ayant exactement une extrémité dans V_2. Comme $V_2 = V \setminus W$, on a alors $A_\Phi(\overline{W}) \neq \emptyset$ et $\delta_\Phi(W) = \emptyset$.

Soit L' le sous-ensemble de chemins de L induit par les demandes de $A_\Phi(\overline{W})$. Le sous-ensemble V_1 est alors composé des sommets v de V' tels que pour chaque arc a de $\delta^{\text{out}}_{D'}(v)$, il existe un arc a' de $\delta^{\text{in}}_{D'}(v)$ avec $a' \prec_{L'} a$. D'après (4.22), on obtient alors $y_a - x_a(A_\Phi(\overline{W}) \cap K^a) = 0$ pour tous les arcs a de $\delta^{\text{out}}_D(W)$. Par ailleurs, comme les chemins de L sont deux-à-deux arc-disjoints, il est facile de voir que $x^k_a = 0$ pour tout arc a de $\delta^{\text{out}}_D(W)$ et pour toute demande k de $A_\Phi(W) \cap K^a$. Par conséquent, la contrainte de vulnérabilité (4.21) associée à W est violée par (x, y). □

En utilisant les propositions 4.19, 4.20 et 4.21, on peut maintenant définir l'ensemble des solutions réalisables pour le 1-PRLPA unitaire.

Théorème 4.22 *L'ensemble* $\{(x, y) \in \{0, 1\}^{m+m_K} : (x, y)$ *satisfait* $(4.10)-(4.19), (4.21)\}$ *correspond à l'ensemble* $S_U(D, v_0, K)$ *des solutions réalisables du 1-PRLPA unitaire.*□

Le théorème précédent permet de formuler le 1-PRLPA unitaire à l'aide du programme linéaire en nombres entiers suivant

$$\min\{c^T y \mid (x, y) \in \{0, 1\}^{m+m_K} : (x, y) \text{ satisfait } (4.10) - (4.19), (4.21)\}. \tag{4.23}$$

Même si seuls les coûts associés aux arcs du véhicule sont considérés dans la fonction objective, on remarque que l'on peut facilement étendre cette dernière de manière à considérer également les coûts associés aux chemins des demandes. De plus, si les nouveaux coûts considérés sont strictement positifs, les contraintes (4.13) et (4.14) ne sont plus nécessaires dans la formulation. Cependant, comme on le montre dans le théorème suivant, l'algorithme de séparation des contraintes de vulnérabilité (4.21) s'appuie sur le fait que le graphe induit par les variables x^k pour tout $k \in K$ est connexe. Comme cette connexité est assurée par les contraintes (4.13) et (4.14), il s'ensuit que l'on doit dans tous les cas les considérer même si elles ne sont plus nécessaires dans la formulation. Par ailleurs, il est facile de voir que si les coûts associés aux arcs empruntés par le véhicule sont strictement positifs, il n'est pas nécessaire non plus de considérer les contraintes (4.10) dans la formulation (4.23). En effet, aucune solution optimale ne contient dans ce cas de circuit pour le véhicule avec aucune demande transportée sur l'un de ses arcs. Pour les autres circuits, la connexité est assurée par les contraintes de vulnérabilité (4.21).

On s'intéresse maintenant à la complexité de la relaxation linéaire de (4.23). On introduit d'abord la notation suivante qui sera utilisée dans le lemme suivant donnant la complexité de la séparation des contraintes de vulnérabilité (4.21). Soient x un vecteur de $\mathbb{R}^{m_K}$ et k une demande de K. On note par D^k le graphe associé à x et k, correspondant au graphe induit par l'ensemble d'arcs traversés par la demande k. Autrement dit, le graphe D^k est induit par l'ensemble d'arcs $\{a \in A^k : x^k_a > 0\}$.

Lemme 4.23 *Soit $(\bar{x}, \bar{y})$ un vecteur de $\mathbb{R}^{m_K}_+ \times \mathbb{R}^m$. Si le graphe D^k associé à $\bar{x}$ et k est connexe pour toute demande $k \in K$, alors le problème de séparation des contraintes de vulnérabilité* (4.21) *peut être résolu en temps polynomial.*

Preuve. On montre que la séparation des contraintes de vulnérabilité se ramène à un nombre polynomial de calculs de coupe minimum dans un graphe auxiliaire. Soit $\hat{D} = (\hat{V}, \hat{A})$ le graphe obtenu à partir de $D = (V, A)$ en contractant les sommets o^k et d^k en un sommet v^k pour toute demande $k \in K$. Soit $\hat{w} \in \mathbb{R}^{\hat{A}}$ le vecteur associé aux arcs de $\hat{A}$ tel que pour tout $(u, v) \in \hat{A}$,

$$\hat{w}_{(u,v)} = \begin{cases} \beta_{(o^k,o^{k'})} + \beta_{(d^k,o^{k'})} + \beta_{(o^k,d^{k'})} + \beta_{(d^k,d^{k'})} & \text{s'il existe } k \in K \text{ tel que } u = o^k \text{ ou } u = d^k \\ & \text{et } k' \in K \text{ tel que } v = o^{k'} \text{ ou } v = d^{k'}, \\ \beta(o^k, v) + \beta(d^k, v) & \text{s'il existe } k \in K \text{ tel que } u = o^k \text{ ou } u = d^k, \\ \beta(u, o^k) + \beta(u, d^k) & \text{s'il existe } k \in K \text{ tel que } v = o^k \text{ ou } v = d^k, \\ \beta_{(u,v)} & \text{sinon,} \end{cases}$$

où $\beta_{(u,v)}$, $u \neq v \in V$ est donné par

$$\beta_{(u,v)} = \begin{cases} y_{(u,v)} - x_{(u,v)}(K^{(u,v)}) & \text{si } {}_{(u,v)} \in A, \\ 0 & \text{sinon.} \end{cases}$$

Pour tout $(u,v) \in \hat{A}$, la valeur $\hat{w}_{(u,v)}$ correspond à la somme des valeurs $y_a - x_a(K^a)$ associées aux arcs $a \in A$ tels que la contraction des sommets o^k et d^k en v^k pour tout $k \in K$ transforme a en l'arc (u,v) dans $\hat{D}$.

Pour toute demande $k \in K$, on note par V^k l'ensemble des sommets du graphe D^k. On définit alors, pour toute demande $k \in K$, l'ensemble d'arcs

$$B^k = \left\{(v^k, v) : v \in V^k \setminus \{o^k, d^k\}\right\}.$$

Considérons le graphe $\tilde{D} = (\hat{V}, \tilde{A})$ obtenu en rajoutant les arcs de B^k dans $\hat{D}$ pour tout $k \in K$. Autrement dit, $\tilde{A} = \left(\cup_{k\in K} B^k\right) \cup \hat{A}$. Soit $\tilde{w} \in \mathbb{R}^{\tilde{A}}$ le vecteur associé aux arcs du graphe $\tilde{D}$ tel que

$$\tilde{w}_a = \begin{cases} +\infty & \text{si } a \in \cup_{k\in K} B^k, \\ \hat{w}_a & \text{sinon,} \end{cases} \qquad \forall\ a \in \tilde{A}.$$

Considérons une demande de K, disons $\bar{k}$. On montre maintenant qu'une contrainte de vulnérabilité associée à un sous-ensemble W tel que $\bar{k} \in A_\Phi(\overline{W})$ est violée si et seulement s'il existe une $v_0 v^{\bar{k}}$-coupe dont le poids est strictement inférieur à 1 dans le graphe $\tilde{D}$ avec les poids $\tilde{w}$.

Supposons que le vecteur $(\bar{x}, \bar{y})$ viole la contrainte de vulnérabilité associée à W. Supposons également que la demande $\bar{k}$ a ses deux extrémités dans $\overline{W}$. On a alors

$$\sum_{a\in\delta^{\text{out}}(W)} \bar{y}_a - \sum_{k\in A_\Phi(\overline{W})} \sum_{a\in\delta^{\text{out}}(W)\cap A^k} \bar{x}_a^k + M \sum_{k\in A_\Phi(W)} \sum_{a\in\delta^{\text{out}}(W)\cap A^k} \bar{x}_a^k < 1.$$

Comme $\bar{x}$ est non négatif, cela implique que pour toute demande $k \in A_\Phi(W)$, $\bar{x}_a^k = 0$ pour tout arc $a \in \delta^{\text{out}}(W) \cap A^k$. Soit W' le sous-ensemble obtenu en contractant l'origine o^k et la destination d^k en un sommet v^k pour toute demande $k \in K$. Il est clair que $v_0 \in W'$ et $v^{\bar{k}} \in \overline{W}'$. Comme $\delta_\Phi(W) = \emptyset$, par la définition de $\hat{w}$ on a

$$\hat{w}(\delta^{\text{out}}(W')) = \sum_{a\in\delta^{\text{out}}(W)} \bar{y}_a - \sum_{k\in A_\Phi(\overline{W})} \sum_{a\in\delta^{\text{out}}(W)\cap A^k} \bar{x}_a^k.$$

Comme les graphes D^k, $k \in K$, sont connexes et pour toute demande de $A_\Phi(W)$, aucun arc de D^k ne traverse la coupe $\delta^{\text{out}}(W)$, tous les sommets de D^k appartiennent à W pour toute demande de $A_\Phi(W)$. Par conséquent, il n'existe aucun arc de B^k, $k \in K$,

appartenant à la coupe $\delta^{\text{out}}(W')$. On déduit alors que $\hat{w}(\delta^{\text{out}}(W')) = \tilde{w}(\delta^{\text{out}}(W')) < 1$. La coupe associée à W' a un poids strictement inférieur à 1 dans le graphe $\tilde{D}$. Finalement, comme $v_0 \in W'$ et $v^{\bar{k}} \in \overline{W}'$, il existe alors une $v_0 v^{\bar{k}}$-coupe dont le poids est strictement inférieur à 1 dans $\tilde{D}$ avec les poids $\tilde{w}$. La preuve de la réciproque est similaire.

La recherche d'une contrainte violée associée à un sous-ensemble W tel que $\bar{k}$ appartient à $A_\Phi(\overline{W})$ se ramène alors à déterminer une $v_0 v^{\bar{k}}$-coupe minimum dans $\tilde{D}$ avec les poids $\tilde{w}$. Comme $\bar{k}$ est une demande quelconque de K telle que $v_0 \notin \{o^k, d^k\}$, cela signifie que la séparation des contraintes de vulnérabilité (4.21) se ramène au calcul d'au plus p coupes minimums dans le graphe $\tilde{D}$. Comme la construction de $\tilde{D}$ à partir de D et le calcul d'une coupe minimum peuvent se réaliser en temps polynomial, la séparation des contraintes de vulnérabilité peut elle aussi être réalisée en temps polynomial. □

Ce lemme permet de montrer que la relaxation linéaire de (4.23) peut être résolue en temps polynomial.

Théorème 4.24 *La relaxation linéaire de* (4.23) *peut être résolue en temps polynomial.*

Preuve. Comme les contraintes (4.11)-(4.13), (4.15)-(4.19) sont en nombre polynomial dans (4.23), la complexité de la relaxation linéaire dépend uniquement de la complexité du problème de séparation des inégalités (4.10), (4.14) et (4.21). Notons par $(\bar{x}, \bar{y})$ la solution à séparer.
La séparation des contraintes (4.10) peut se ramener au calcul de $|A|$ $v_0 v^a$-coupes minimum, $a \in A$, où v^a est le sommet obtenu de la contraction de a, la fonction poids étant donnée par $\bar{y} \geq 0$. Ce problème peut être résolu en temps polynomial. De la même manière, pour toute demande $k \in K$, la séparation des contraintes (4.14) associées à k se ramène au calcul de $|V|$ $v^k v$-coupes minimum, où v^k est le sommet obtenu par la contraction des sommets o^k et d^k, $k \in K$, et v un sommet du graphe ; la fonction poids sur les arcs étant donnée par $\bar{x}^k \geq 0$.
Supposons maintenant que $(\bar{x}, \bar{y})$ vérifie les contraintes de connexité des demandes (4.14). Dans ce cas, pour toute demande $k \in K$, le graphe D^k associé à $\bar{x}$ et k est connexe. De plus, par les contraintes triviales (4.18), le vecteur $\bar{x}$ est non négatif. D'après le lemme 4.23, la séparation des contraintes de vulnérabilité (4.21) par rapport à $(\bar{x}, \bar{y})$ est donc polynomiale. □

On remarque que les contraintes de vulnérabilité (4.21) sont dominées par les contraintes suivantes.

$$\sum_{a\in\delta^{\text{out}}(W)} y_a - \sum_{k\in A_\Phi(\overline{W})} \sum_{a\in\delta^{\text{out}}(W)\cap A^k} x_a^k \geq 1, \tag{4.24}$$

pour tout $W \subset V$ tel que $v_0 \in W$, $A_\Phi(\overline{W}) \neq \emptyset$ et $\delta_\Phi(W) = \emptyset$. Ces contraintes, appelées *contraintes de vulnérabilité renforcées*, suffisent, avec les contraintes (4.10)-(4.19), à formuler le 1-PRLPA unitaire sous forme d'un programme linéaire en nombres entiers. Cependant, la complexité du problème de séparation associé aux contraintes (4.24) est toujours une question ouverte. Néanmoins, ce problème est polynomial si x est entier. En effet, le problème de séparation des contraintes (4.24) n'est rien d'autre dans ce cas que le problème de séparation des contraintes de vulnérabilité (4.21).

4.3 Formulation du 1-PRLPA

On s'intéresse maintenant au cas général du 1-PRLPA. L'unique hypothèse que l'on considère sur le volume des demandes est que la capacité du véhicule soit supérieure ou égale au volume de chaque demande. On a montré dans la section 4.1, qu'une représentation minimale des solutions du 1-PRLPA était donnée par la séquence des arcs du trajet du véhicule et par l'ensemble des arcs des chemins des demandes. Pour le 1-PRLPA unitaire, on a présenté dans la section précédente une formulation basée uniquement sur les ensembles d'arcs du trajet du véhicule et des chemins des demandes. Dans cette partie, on étend cette formulation en adaptant les inégalités (4.10)-(4.19),(4.21) pour des volumes des demandes et une capacité du véhicule quelconques et en ajoutant de nouvelles variables et contraintes pour coder l'ordre des arcs du véhicule.

On considère les ensembles de variables $x \in \{0,1\}^{m_K}$ et $y \in \{0,1\}^m$ définis précédemment. Il est clair que les contraintes (4.10)-(4.14) et les contraintes triviales (4.16)-(4.19) restent valides quel que soit le volume des demandes et la capacité du véhicule. Les seules contraintes qui doivent être généralisées sont les contraintes impliquant à la fois les variables x et y. Les contraintes de capacité (4.15) se généralisent selon

$$Qy_a - \sum_{k\in K^a} q^k x_a^k \geq 0, \tag{4.25}$$

pour tout arc $a \in A$. Les contraintes de vulnérabilité (4.21) ont été introduites dans la formulation (4.23) de façon à garantir que le circuit du véhicule induit par y respecte les contraintes de précédence induites par K. Comme pour le 1-PRLPA général, on considère des variables codant l'ordre dans lequel les arcs du circuit du véhicule sont

traversés, le respect des contraintes de précédence induites par les chemins des demandes de K pourra se faire directement entre ces nouvelles variables et les variables x. Les contraintes de vulnérabilité (4.21) ne sont donc plus nécessaires dans la formulation. On peut cependant généraliser les contraintes de vulnérabilité (4.21) et les contraintes de vulnérabilité renforcées (4.24) afin de renforcer la relaxation linéaire.

Proposition 4.25 *Soit $W \neq \emptyset$ un sous-ensemble propre de sommets de V tel que $v_0 \in W$ et $A_\Phi(\overline{W}) \neq \emptyset$. La contrainte de vulnérabilité généralisée associée à W*

$$\sum_{a \in \delta^{\text{out}}(W)} y_a - \left\lceil \frac{1}{Q} \sum_{k \in A_\Phi(\overline{W})} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} q^k x_a^k \right\rceil \geq 1, \tag{4.26}$$

est valide pour le 1-PRLPA.

Preuve. Soit $K(\overline{W})$ l'ensemble des demandes $k \in A_\Phi(\overline{W})$ tel qu'il existe un arc $a \in \delta^{\text{out}}(W)$ avec $x_a^k > 0$. Le second terme dans le membre de gauche représente le nombre minimum de passages de W à $\overline{W}$ que le véhicule doit effectuer pour transporter les demandes de $K(\overline{W})$ sur les arcs sortants de W. Puisque v_0 appartient à W et que toutes les demandes $K(\overline{W})$ ont leur origine dans $\overline{W}$, le véhicule doit traverser la coupe $\delta^{\text{out}}(W)$ au moins une fois sans transporter aucune demande de $K(\overline{W})$. □

On remarque que l'on ne se restreint pas à considérer uniquement des ensembles W tels que $\delta_\Phi(W) = \emptyset$. En effet, contraitement au cas unitaire, le fait d'avoir $\delta_\Phi(W) \neq \emptyset$ n'implique pas forcément que la contrainte de vulnérabilité (4.26) n'est pas violée par (x, y). Par ailleurs, on aurait pu généraliser les contraintes de vulnérabilité (4.21) comme suit

$$\sum_{a \in \delta^{\text{out}}(W)} y_a - \left\lceil \frac{1}{Q} \sum_{k \in A_\Phi(\overline{W})} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} q^k x_a^k \right\rceil + M \sum_{k \in A_\Phi(W)} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x_a^k \geq 1,$$

où M correspond à une constante suffisamment grande. Ces dernières inégalités sont clairement dominées par les contraintes (4.26).

On s'intéresse maintenant à la modélisation de la séquence des arcs du circuit du véhicule. Une façon naturelle de représenter l'ordre sur cet ensemble d'arcs consiste à considérer les variables d'ordre linéaire [81]

$$\eta_{aa'} = \begin{cases} 1 & \text{si } a \text{ précède } a' \text{ dans le circuit du véhicule,} \\ 0 & \text{sinon,} \end{cases}$$

pour tout couple d'arcs distincts $a, a' \in A$. Comme le vecteur η définit un ordre total sur les arcs traversés par le véhicule, c'est-à-dire, sur l'ensemble d'arcs $A_y = \{a \in A : y_a = 1\}$, les variables (y, η) peuvent être vues comme un ordre linéaire partiel sur A. Ce concept d'ordre linéaire partiel a été introduit par Kerivin et Sirdey [94]. Les variables (y, η) doivent alors satisfaire les contraintes suivantes

$$y_a + y_{a'} - \eta_{aa'} - \eta_{a'a} \leq 1 \qquad \forall\ a \neq a' \in A, \tag{4.27}$$

$$\eta_{aa'} + \eta_{a'a} - y_a \leq 0 \qquad \forall\ a \neq a' \in A, \tag{4.28}$$

$$\eta_{aa'} + \eta_{a'a''} - \eta_{aa''} - y_{a'} \leq 0 \qquad \forall\ a \neq a' \neq a'' \neq a \in A, \tag{4.29}$$

$$\eta_{aa'} \geq 0 \qquad \forall\ a \neq a' \in A, \tag{4.30}$$

$$\eta_{aa'} \leq 1 \qquad \forall\ a \neq a' \in A, \tag{4.31}$$

qui ont été données pour le polytope des ordres linéaires partiels [94].

L'ordre linéaire partiel induit par (y, η) peut cependant ne pas correspondre à un circuit. On doit ajouter de nouvelles contraintes afin d'avoir, pour chaque sommet, une alternance entre arcs entrants et sortants. On doit également imposer que pour tout sommet $v \in V \setminus \{v_0\}$ (respectivement le sommet v_0), le premier arc incident à v (respectivement v_0) dans la séquence soit un arc entrant (respectivement sortant). Ceci peut être fait par l'introduction des contraintes suivantes

$$\sum_{a \in \delta^{\text{out}}(v) \setminus \{a'\}} \eta_{aa'} - \sum_{a \in \delta^{\text{in}}(v)} \eta_{aa'} + y_{a'} = 0 \tag{4.32}$$

pour tout sommet $v \in V \setminus \{v_0\}$ et pour tout arc $a' \in \delta^{\text{out}}(v)$. On définit les contraintes similaires associées au dépôt v_0 comme suit

$$\sum_{a \in \delta^{\text{out}}(v_0) \setminus \{a'\}} \eta_{aa'} - \sum_{a \in \delta^{\text{in}}(v_0)} \eta_{aa'} = 0, \tag{4.33}$$

pour tout arc $a' \in \delta^{\text{out}}(v_0)$. Les contraintes (4.32) sont appelées *contraintes alternées*, et les contraints (4.33) *contraintes alternées du dépôt*. Les contraintes (4.32) impliquent que toute extrémité initiale d'un arc $a \in A_y$ différente du dépôt doit posséder, dans la séquence d'arcs précédant a, un arc entrant de plus que d'arcs sortants. De manière similaire, les contraintes (4.33) impliquent que le nombre d'arcs entrants dans v_0 doit être égal au nombre d'arcs sortants avant qu'un arc sortant de v_0 soit considéré.

Notons par S_{CW} l'ensemble des vecteurs binaires (y, η) induisant des circuits dans D commençant en v_0. On a alors le résultat suivant.

Proposition 4.26 *L'ensemble S_{CW} est défini par*

$$S_{CW} = \{(y, \eta) \in \{0,1\}^{m^2} : (y, \eta) \textit{ satisfait } (4.11), (4.16), (4.17), (4.27)\text{-}(4.33)\}$$

Preuve. Considérons un circuit potentiellement non élémentaire C de D commençant en v_0. Soit (y, η) le vecteur tel que $y_a = 1$ si et seulement si $a \in C$ et $\eta_{aa'} = 1$ si et seulement si a apparaît avant a' dans C. Il est facile de voir que dans ce cas, (y, η) satisfait les contraintes (4.11),(4.16),(4.17),(4.27)-(4.33).

On prouve maintenant que tout vecteur binaire (y, η) satisfaisant les contraintes (4.11),(4.16),(4.17),(4.27)-(4.33) induit un circuit dans D commençant en v_0. Comme nous l'avons déjà mentionné, η correspond à un ordre linéaire partiel sur l'ensemble d'arcs A_y. Les contraintes (4.32) impliquent que pout tout sommet $v \in V \setminus \{v_0\}$, le premier arc incident à v est un arc entrant. On en déduit alors que le premier arc de l'ordre total, disons a_1, est un arc dont l'extrémité initiale est v_0. À cause des contraintes (4.32) et (4.33), le second arc dans la séquence doit sortir de l'extrémité terminale de a_1. En répétant cet argument, il est clair que (y, η) correspond correspond à un chemin commençant en v_0. De plus, comme y satisfait les contraintes (4.11), ce chemin est un circuit. Ce dernier résultat termine la preuve. □

Les contraintes de conservation de flot (4.11) associées aux sommets $v \in V \setminus \{v_0\}$ ne sont pas nécessaires dans la description de l'ensemble S_{CW}. Seule celle associée au dépôt v_0 doit être prise en compte. De même, les contraintes de connexité (4.10) n'apparaissent pas dans la description de S_{CW} car on considère maintenant un ordre linéaire total η sur A_y. En fait, ce dernier assure que le graphe induit par l'ensemble A_y est faiblement connexe.

Pour obtenir l'ensemble des solutions réalisables du 1-PRLPA, que l'on note par $S_G(D, v_0, K)$, on doit maintenant combiner le circuit du véhicule et les chemins élémentaires des demandes de manière à ce que le circuit respecte les contraintes de précédence induites par les chemins des demandes. Autrement dit, l'ordre défini par (y, η) doit être conforme aux chemins associés aux demandes de K. Pour cela, on introduit les *contraintes de précédence des demandes*

$$x_a^k + x_{a'}^k - \eta_{aa'} \leq 1 \tag{4.34}$$

pour tout $k \in K$, pour tout $v \in V \setminus \{o^k, d^k\}$, pour tout $a \in \delta^{\text{in}}(v) \cap A^k$ et pour tout $a' \in \delta^{\text{out}}(v) \cap A^k$. Ces contraintes permettent de transporter chaque demande $k \in K$ d'un sommet $v \in V \setminus \{o^k, d^k\}$ sur un arc sortant $a \in \delta^{\text{out}}(v) \cap A^k$ si et seulement si l'arc $a' \in \delta^{\text{in}}(v) \cap A^k$ sur lequel la demande k est transportée jusqu'en v apparait avant a dans le circuit. Il est clair que les inégalités (4.34) sont valides pour le 1-PRLPA. De plus, d'après les contraintes (4.13), les demandes sont transportées sur des chemins élémentaires et traversent donc chaque sommet au maximum une fois. Les contraintes (4.34) suffisent donc pour s'assurer que le circuit satisfait bien les

contraintes de précédence induites par les demandes. On peut alors définir l'ensemble $S_G(D, v_0, K)$ des vecteurs binaires (x, y, η) associées aux solutions réalisables du 1-PRLPA. En utilisant les précédents résultats, on donne la description de $S_G(D, v_0, K)$ dans le théorème suivant.

Théorème 4.27 *L'ensemble $S_G(D, v_0, K)$ des solutions réalisables du 1-PRLPA est $S_G(D, v_0, K) = \{(x, y, \eta) \in \{0, 1\}^{m^2+m_K} : (x, y, \eta)$ satisfait (4.11)-(4.14), (4.16)-(4.19), (4.25), (4.27)-(4.34)$\}$.* □

On remarque que les contraintes (4.32) et (4.33) peuvent être remplacées par les deux ensembles de contraintes suivants

$$\sum_{a \in \delta^{\text{in}}(v) \setminus \{a'\}} \eta_{aa'} - \sum_{a \in \delta^{\text{out}}(v)} \eta_{aa'} = 0 \qquad \forall\ a' \in \delta^{\text{in}}(v), \tag{4.35}$$

$$\sum_{a \in \delta^{\text{in}}(v_0) \setminus \{a'\}} \eta_{aa'} - \sum_{a \in \delta^{\text{out}}(v_0)} \eta_{aa'} + y_{a'} = 0 \quad \forall\ a' \in \delta^{\text{in}}(v_0). \tag{4.36}$$

Ces contraintes sont clairement valides pour le 1-PRLPA. La preuve de suffisance des contraintes (4.35) et (4.36) est similaire à celle des contraintes (4.32) et (4.33). Cependant, l'argumentation doit commencer avec le dernier arc de la séquence et regarder à chaque itération l'arc précédent dans la séquence.

En utilisant le précédent théorème, on peut maintenant formuler le 1-PRLPA à l'aide du programme linéaire en nombres entiers (4.37) suivant

$$\begin{aligned} &\min\{c^T y \mid (x, y, \eta) \in \{0, 1\}^{m^2+m_K} : (x, y, \eta) \text{ satisfait (4.11)-(4.14),} \\ &\text{(4.16)-(4.19), (4.25), (4.27)-(4.34)}\}. \end{aligned} \tag{4.37}$$

Les remarques concernant la fonction objective et les contraintes de circuit (4.13) énoncées pour (4.23) s'appliquent également pour (4.37). On remarque également que ce dernier contient un nombre polynomial de contraintes et de variables, ce qui nous permet d'énoncer le théorème suivant.

Théorème 4.28 *La relaxation linéaire de la formulation* (4.37) *peut être résolue en temps polynomial.* □

Contrairement à la formulation que l'on a donnée pour le 1-PRLPA unitaire, (4.37) ne contient aucune contrainte de vulnérabilité. Cependant, comme nous l'avons déjà fait remarquer, considérer les contraintes (4.26), ou quelques variantes, peut renforcer la relaxation linéaire de la formulation. Malheureusement, la complexité du problème

de séparation de ces contraintes n'a pas encore été prouvée, et nous conjecturons que ce problème est NP-difficile.

Dans ce chapitre, nous nous sommes intéressés au problème de ramassage et livraison mono-véhicule préemptif asymétrique ainsi qu'à la version unitaire du problème lorsque le véhicule ne transporte qu'une demande à la fois. Pour ces deux versions, nous avons déterminé l'information minimale correspondant à l'information minimum nécessaire pour déterminer en temps polynomial si une solution est réalisable. En utilisant ces résultats, nous avons donné pour ces deux versions deux formulations à l'aide de programmes linéaires en nombres entiers. Dans les chapitres suivants, nous concentrons notre étude sur le 1-PRLPA unitaire. Nous étudions le polytope des solutions de ce problème. Nous donnons également une heuristique valide pour le problème et développons un algorithme de coupes et branchements, basé sur la formulation du problème introduite dans ce chapitre, pour résoudre ce problème.

Chapitre 5

Polyèdre associé au 1-PRLPA unitaire

Dans ce chapitre, on étudie l'enveloppe convexe des solutions du problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire. On caractérise sa dimension et l'on donne des conditions nécessaires et suffisantes pour que les contraintes de la formulation (4.23) définissent des facettes. Par la suite, on développe un algorithme de coupes et branchements pour résoudre le 1-PRLPA unitaire et l'on discute des résultats obtenus. Finalement, on considère le problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire lorsque le graphe est un circuit doublement orienté et chaque sommet différent du dépôt est incident à au moins une demande. On donne dans ce cas, la description polyédrale du polytope des solutions du 1-PRLPA unitaire.

5.1 Étude polyédrale du 1-PRLPA unitaire

Dans cette section, on caractérise la dimension du polytope des solutions du 1-PRLPA unitaire et l'on donne des conditions nécessaires et suffisantes pour que les contraintes de la formulation (4.23) définissent des facettes.

Comme dans le chapitre précédent, une instance du 1-PRLPA unitaire est représentée par le triplet (D, v_0, K). Dans cette étude polyédrale, on suppose que le graphe D est complet et chaque sommet du graphe excepté le dépôt est incident à au plus une demande, le dépôt étant incident à aucune demande. Ces hypothèses ne sont pas restrictives. En effet, toute instance peut être modifiée, en créant des copies de sommets et en ajoutant des arcs fictifs de coût infinis, afin que ces hypothèses soient vérifiées.

De plus, nous considérons que l'ensemble K contient au minimum deux demandes. En effet, le problème avec une seule demande n'a pas d'intérêt du point de vue des rechargements. Nous avons alors $p \geq 2$ et $n \geq 5$.

Dans ce qui suit, toute solution du 1-PRLPA unitaire sera représentée par une paire (L, T) avec $L = \{L_1, \ldots, L_p\}$, où L_k est le chemin de la demande k, $k = 1, 2, \ldots, p$ et T est l'ensemble des arcs traversés par le véhicule. Étant donnée une solution (L, T), on notera par (x^L, y^T) son vecteur d'incidence. Par souci de concision, on utilisera parfois des arcs fictifs de la forme (v, v). Comme par hypothèse, notre graphe ne contient pas de boucle, chaque ensemble d'arcs sera toujours identifié par ses arcs non fictifs.

Dans la partie suivante, on caractérise la dimension du polytope du 1-PRLPA unitaire. Pour cela, on présente maintenant quelques résultats sur les solutions du 1-PRLPA unitaire qui seront utilisés par la suite.

Proposition 5.1 *Soit T un sous-ensemble d'arcs de A et $L = \{L_1, L_2, \ldots, L_p\}$ correspondant à un ensemble de $o^k d^k$-chemins, $k \in K$, utilisant seulement des arcs de T. Supposons que le graphe orienté induit par T est Eulérien et que les chemins des demandes L_k, $k \in K$ sont arc-disjoints. Supposons également que T contient une v_0-arborescence Γ qui couvre toutes les origines de P. Supposons qu'au maximum un chemin de L, disons $L_{\bar{k}}$, intersecte Γ et qu'il existe un chemin de v_0 à $o^{\bar{k}}$ dans $\Gamma \setminus L_{\bar{k}}$. Alors la solution (L, T) est réalisable pour le 1-PRLPA unitaire.*

Preuve. Considérons le vecteur d'incidence (x^L, y^T) de (L, T) avec

$$x_a^k = \begin{cases} 1 & \text{si } a \in L_k, \\ 0 & \text{sinon,} \end{cases}$$

pour $k \in K$ et,

$$y_a = \begin{cases} 1 & \text{si } a \in T, \\ 0 & \text{sinon.} \end{cases}$$

On prouve que le vecteur (x^L, y^T) satisfait les contraintes de (4.23). En effet, comme T induit un graphe Eulérien et par définition de L, il est clair que les contraintes (4.10)-(4.12) sont satisfaites. De plus, comme les chemins L_k, $k \in K$, sont arc-disjoints et utilisent seulement des arcs de T, les inégalités (4.15) sont aussi satisfaites. Considérons maintenant une contrainte de vulnérabilité induite par un sous-ensemble de sommets W contenant v_0 et tel que $A_\Phi(\overline{W}) \neq \emptyset$ et $\delta_\Phi(W) = \emptyset$. Remarquons d'abord que, puisque $A_\Phi(\overline{W}) \neq \emptyset$, il doit exister au moins une origine dans $\overline{W}$. Par ailleurs, comme $v_0 \in W$ et Γ couvre toutes les origines, nous devons avoir $\delta^{\text{out}}(W) \cap \Gamma \neq \emptyset$. On distingue

maintenant deux cas. Supposons qu'il existe un arc $\tilde{a}$ de $\delta^{\text{out}}(W) \cap \Gamma$ qui n'intersecte pas $L_{\bar{k}}$. Comme Γ n'intersecte aucun autre chemin d'une demande, nous avons alors $y_{\tilde{a}} - x_{\tilde{a}}(K^{\tilde{a}}) = 1$. Par les contraintes de capacité, on a également $y_a - x_a(K^a) \geq 0$ pour tout arc $a \in \delta^{\text{out}}(W)$. Ceci implique que $y(\delta^{\text{out}}(W)) - \sum_{k \in K} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x_a^k \geq 1$. La contrainte de vulnérabilité associée à W est donc satisfaite par (x, y). Si tous les arcs de $\delta^{\text{out}}(W) \cap \Gamma$ appartiennent à $L_{\bar{k}}$, alors $\bar{k}$ a ses deux extrémités dans W. Autrement, d'après la définition de W, cela implique que $\bar{k}$ a son origine $o^{\bar{k}}$ dans $\overline{W}$ et cela contredit alors l'hypothèse de l'existence d'un chemin de v_0 à $o^{\bar{k}}$ dans $\Gamma \setminus L_{\bar{k}}$. Considérons un arc $\tilde{a}$ de $\delta^{\text{out}}(W) \cap \Gamma$. Comme les chemins des demandes sont arc-disjoints, on a $y_{\tilde{a}} - \sum_{k \in A_\Phi(\overline{W}) \cap K^{\tilde{a}}} x_{\tilde{a}}^k = 1$ ce qui signifie que la contrainte de vulnérabilité associée à W est satisfaite par (x, y). □

Proposition 5.2 *Étant donné un sommet $v \in V$, il existe une solution (L, T) du 1-PRLPA unitaire avec $T = \delta(v)$.*

Preuve. Nous montrons que la solution (L, T) définie par $L = \{L_1, \ldots, L_p\}$ avec $L_k = \{(o^k, v), (v, d^k)\}$ pour tout $k = 1, 2, \ldots, p$ et $T = \delta(v)$ est réalisable pour le 1-PRLPA unitaire. En effet, T induit un graphe Eulérien et les chemins des demandes L_k, $k \in K$ utilisent uniquement des arcs de T et sont arc-disjoints.Considérons l'ensemble d'arcs $\Gamma = \{(v_0, v)\} \cup \{(v, o^k) : k \in K\}$. Γ est une v_0-arborescence qui couvre toutes les origines et n'intersecte aucun chemin L_k, $k \in K$. Par conséquent, d'après la proposition 5.1, la solution (L, T) est réalisable pour le 1-PRLPA unitaire. □

Soit $P_U(D, v_0, K)$ l'enveloppe convexe des solutions du programme linéaire en nombres entiers (4.23), *i.e.*,

$$P_U(D, v_0, K) = \text{conv}\{(x, y) \in \{0, 1\}^{m+m_K} : (x, y) \text{ satisfait } (4.10) - (4.21)\}.$$

Dans les sous-sections suivantes, nous étudions ce polyèdre.

5.1.1 Dimension de $P_U(D, v_0, K)$

Dans cette section, nous caractérisons la dimension de $P_U(D, v_0, K)$. Pour cela, nous montrons d'abord que toute équation de $P_U(D, v_0, K)$ est une combinaison linéaire des contraintes (4.11) et (4.12).

Proposition 5.3 *Toute équation $r^T x + s^T y = \beta$ de $P_U(D, v_0, K)$ est une combinaison linéaire des équations* (4.11) *et* (4.12).

Preuve. Soit $r^T x + s^T y = \beta$ une équation de $P_U(D, v_0, K)$. On montre qu'il existe $\lambda = (\lambda_v^k, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = (\lambda, \mu)M_{CF}$, où M_{CF} est la matrice des contraintes de conservation de flots.

Soit v_1 un sommet de V. La proposition 5.2 implique qu'il existe une solution, disons (L^1, T^1), avec $T^1 = \delta(v_1)$. Soit u, v et w trois sommets distincts de $V \backslash \{v_1\}$. Les solutions (L^1, T^2) et (L^1, T^3), avec $T^2 = T^1 \cup \{(u, v), (v, u)\}$ et $T^3 = T^1 \cup \{(u, v), (v, w), (w, u)\}$ sont également réalisables pour $P_U(D, v_0, K)$. Cela implique que les vecteurs d'incidence de ces solutions vérifient l'équation $r^T x + s^T y = \beta$. On a alors $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^2} = x^{L^1} + y^{T^3}$. Comme u, v et w sont des sommets quelconques de $V \setminus \{v_1\}$, il s'ensuit que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V \setminus \{v_1\}, \tag{5.1}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0, \quad \forall\ u \neq v \neq w \neq u \in V \setminus \{v_1\}. \tag{5.2}$$

Comme le sommet v_1 est un sommet quelconque de V, il est possible, en réitérant ce raisonnement pour trois autres sommets différents de v_1, d'étendre les équations (5.1) et (5.2) comme suit

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V, \tag{5.3}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0, \quad \forall\ u \neq v \neq w \neq u \in V. \tag{5.4}$$

Posons $\mu_{v_0} = 0$ et $\mu_v = -s_{(v_0,v)}$ pour tout $v \in V \setminus \{v_0\}$. On a alors $s_{(v_0,v)} = \mu_{v_0} - \mu_v$ pour tout $v \in V \setminus \{v_0\}$. Les équations (5.3) impliquent que $s_{(v,v_0)} = \mu_v - \mu_{v_0}$ pour tout $v \in V \setminus \{v_0\}$. Considérons deux sommets distincts $u \neq v$ de $V \setminus \{v_0\}$. En considérant l'équation (5.4) associée aux sommets v_0, u et v, on obtient $s_{(v_0,u)} + s_{(u,v)} + s_{(v,v_0)} = 0$. D'après les valeurs de s associées aux arcs incidents à v_0, on déduit que $s_{(u,v)} = \mu_u - \mu_v$. Comme u et v sont choisis arbitrairement, on a alors

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u, v) \in A. \tag{5.5}$$

On déduit alors que $s = \mu M_y$, où M_y est la matrice des contraintes de conservation de flot (4.11).

Considérons maintenant le vecteur r associé à une demande de K, disons $\bar{k}$. Soit (u, v) un arc de $A^{\bar{k}}$ n'appartenant pas à $\delta^{\text{out}}(o^{\bar{k}})$. On suppose, sans perte de généralité, qu'il existe une demande de $K \setminus \{\bar{k}\}$, disons $\tilde{k}$, ayant pour origine le sommet u et pour destination le sommet v. Soit L^2 un ensemble de p chemins élémentaires défini par

$$L_k^2 = \begin{cases} \{(o^k, u), (u, v), (v, d^k)\} & \text{si } k = \bar{k}, \\ \{(o^k, v_0), (v_0, d^k)\} & \text{si } k = \tilde{k}, \\ \{(o^k, d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Le chemin L^2_k correspond à un chemin élémentaire de o^k à d^k pour toute demande $k \in K$. De plus, L^2 est un ensemble de chemins arc-disjoints. Considérons la solution (L^2, A). Il est facile de voir que cette solution vérifie les contraintes (4.10)-(4.19). De plus, par construction, on a $\sum_{k \in A_\Phi(\overline{W})} \sum_{a \in \delta^{\text{out}}(W) \cap A^k} x^k_a \leq 1$ pour tout sous-ensemble de sommets $W \subset V$ avec $v_0 \in W$, $A_\Phi(\overline{W}) \neq \emptyset$ et $\delta_\Phi(W) = \emptyset$. Comme on a $y(\delta^{\text{out}}(W)) \geq 2$ pour tout $\emptyset \neq W \subset V$, (x, y) vérifie donc les contraintes de vulnérabilité renforcées (4.24). Par conséquent, (x, y) est réalisable pour $P_U(D, v_0, K)$.

On peut remarquer que l'arc $(o^{\bar{k}}, v)$ n'appartient pas à L^2. On peut donc définir une solution (L^3, A) avec

$$L^3_k = \begin{cases} \{(o^k, v), (v, d^k)\} & \text{si } k = \bar{k}, \\ L^2_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Il est clair que (L^3, A) est également réalisable pour $P_U(D, v_0, K)$. Les deux solutions vérifient alors l'équation $r^T x + s^T y = \beta$, ce qui implique que $x^{L^2} + y^A = x^{L^3} + y^A$. On déduit alors que $r^{\bar{k}}_{(o^{\bar{k}},u)} + r^{\bar{k}}_{(u,v)} = r^{\bar{k}}_{(o^{\bar{k}},v)}$. Comme (u, v) est un arc quelconque de $A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$, on obtient

$$r^{\bar{k}}_{(o^{\bar{k}},u)} + r^{\bar{k}}_{(u,v)} = r^{\bar{k}}_{(o^{\bar{k}},v)} \qquad \forall\ (u, v) \in A^k \setminus \delta^{\text{out}}(o^{\bar{k}}). \tag{5.6}$$

Posons $\lambda^{\bar{k}}_{o^{\bar{k}}} = 0$ et $\lambda^{\bar{k}}_v = -r^{\bar{k}}_{(o^{\bar{k}},v)}$ pour tout $v \in V \setminus \{o^{\bar{k}}\}$. Alors $r^{\bar{k}}_{(o^{\bar{k}},v)} = \lambda^{\bar{k}}_{o^{\bar{k}}} - \lambda^{\bar{k}}_v$ pour tout $v \in V \setminus \{o^{\bar{k}}\}$. Soit (u, v) un arc de $A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$. L'équation (5.6) associée à l'arc (u, v) et les valeurs de $r^{\bar{k}}$ associées aux arcs de $\delta^{\text{out}}(o^{\bar{k}})$ permettent de déduire que $r^{\bar{k}}_{(u,v)} = \lambda^{\bar{k}}_u - \lambda^{\bar{k}}_v$. Comme $\bar{k}$ est une demande quelconque de K et (u, v) est un arc quelconque de $A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$, il s'ensuit que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\ k \in K, \forall\ (u, v) \in A^k. \tag{5.7}$$

Ceci implique que le vecteur r n'est rien d'autre qu'une combinaison linéaire des équations (4.12), ce qui termine la preuve. □

Nous pouvons maintenant donner la dimension du polytope.

Théorème 5.4 *$dim(P_U(D, v_0, K)) = (n-1)(p(n-3) + (n-1)) + p$.*

Preuve. Par la proposition 5.3, les seules équations du polytope $P_U(D, v_0, K)$ sont les contraintes de conservation de flot (4.11) et (4.12). La matrice des équations M_{CF}

de $P_U(D, v_0, K)$ est alors de la forme :

$$M_{CF} = \begin{pmatrix} M_y & & & \\ & M_1 & & \\ & & \ddots & \\ & & & M_p \end{pmatrix}$$

où M_y (resp. M_k, $k \in K$) correspond à la matrice d'adjacence arc-sommet de D (resp. $D(A_k)$, $k \in K$). On a $dim(P_U(D, v_0, K)) = N - \text{rang}(M_{CF})$, où $N = n(n-1)+p(n(n-1)-2(n-1)+1)$ est le nombre de variables. D'un autre côté, la matrice d'adjacence arc-sommet A d'un graphe orienté faiblement connexe D avec n sommets satisfait $\text{rang}(A) = n-1$ [72]. Comme les graphes D et $D(A_k)$, $k \in K$ sont faiblement connexes et contiennent n sommets, par le résultat précédent, on obtient $\text{rang}(M_y) = n-1$ et $\text{rang}(M_k) = n-1$, $k \in K$. Par la structure de M_{CF}, on a alors $\text{rang}(M_{CF}) = (p+1)(n-1)$. □

5.1.2 Étude faciale du polytope du 1-PRLPA unitaire

Dans cette section, nous étudions la structure faciale du polytope $P_U(D, v_0, K)$ du problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire. En particulier, nous donnons des conditions nécessaires et suffisantes pour que les contraintes de la formulation (4.23) du 1-PRLPA unitaire définissent des facettes. Étant donnée une solution (L, T) du 1-PRLPA unitaire et une face F induite par une inégalité valide du 1-PRLPA unitaire telle que le vecteur d'incidence (x^L, y^T) de la solution (L, T) appartient à F, on dit que (L, T) appartient à F. On considère d'abord les contraintes triviales. Avant cela, on donne deux propositions donnant des conditions suffisantes pour l'existence de solutions réalisables pour le 1-PRLPA unitaire.

Proposition 5.5 *Soient T un sous-ensemble d'arcs de A induisant un graphe Eulérien et B un sous-ensemble de T. Soient $\bar{k}$ une demande de K et $L_{\bar{k}}$ un $o^{\bar{k}}d^{\bar{k}}$-chemin élémentaire dans B. Si le graphe induit par $T \setminus B$ est p-arc connexe, alors il existe dans $T \setminus B$ un ensemble $L' = \{L_k : k \in K \setminus \bar{k}\}$ de $o^k d^k$-chemins élémentaires, $k \in K \setminus \{\bar{k}\}$, tel que $(L' \cup \{L_{\bar{k}}\}, T)$ est réalisable pour le 1-PRLPA unitaire.*

Preuve. Notons D' le graphe induit par $T \setminus B$. D'après le corollaire du théorème des branchements disjoints d'Edmonds [90] (Corollaire 53.1a, p. 905), comme D' est p-arc connexe, il existe alors p arborescences disjointes $\Gamma_0, \Gamma_1, \ldots, \Gamma_{p-1}$ dans D'. On associe les $p-1$ dernières arborescences aux demandes de $K \setminus \{\bar{k}\}$. Soit Γ_k l'arborescence associée

à la demande k. On suppose, sans perte de généralité, que Γ_0 (respectivement Γ_k, $k \in K \setminus \{\bar{k}\}$) a pour racine v_0 (respectivement o^k). Pour chaque demande $k \in K \setminus \{\bar{k}\}$, il existe dans Γ_k un chemin élémentaire de o^k à d^k que l'on note L_k. On a alors un ensemble $L = \{L_1, L_2, \ldots, L_p\}$ de $o^k d^k$-chemins élémentaires arc-disjoints puisque les arborescences sont disjointes et n'utilisent aucun arc de B. On a également une v_0-arborescence Γ_0 qui n'intersecte aucun arc de L. D'après la proposition 5.1, (L, T) est donc réalisable pour le 1-PRLPA unitaire. □

Proposition 5.6 *Soit T un ensemble d'arcs de A induisant un graphe Eulérien $(p+1)$-arc connexe. Il existe alors $L = \{L_k : k \in K\}$ tel que (L, T) est une solution réalisable pour le 1-PRLPA unitaire.*

Preuve. Soit D' le graphe induit par T. D'après le corollaire du théorème des branchements disjoints d'Edmonds [90] (Corollaire 53.1a, p. 905), comme D' est $(p+1)$-arc connexe, il existe $p+1$ arborescences disjointes $\Gamma_0, \Gamma_1, \ldots, \Gamma_p$ dans D'. Associons les p dernières arborescences aux demandes de K et notons par Γ_k l'arborescence associée à la demande k. On suppose, sans perte de généralité, que Γ_0 (respectivement Γ_k, $k \in K$) a pour racine v_0 (respectivement o^k). Pour chaque demande $k \in K$, on peut trouver dans Γ_k un chemin élémentaire de o^k à d^k, disons L_k. On a alors un ensemble $L = \{L_1, L_2, \ldots, L_p\}$ de $o^k d^k$-chemins élémentaires arc-disjoints puisque les arborescences sont disjointes. On a également une v_0-arborescence couvrante Γ_0 qui n'intersecte aucun arc de L. D'après la proposition 5.1, (L, T) est donc réalisable pour le 1-PRLPA unitaire. □

5.1.2.1 Inégalités triviales

On remarque d'abord que toute inégalité $y_a \geq 0$ associée à un arc $a \in A$ est dominée par la contrainte de capacité (4.15) associée à a, lorsque au moins une demande peut être transportée sur a, *i.e.*, $K^a \neq \emptyset$. Si aucune demande peut utiliser l'arc a, l'inégalité $y_a \geq 0$ n'est rien d'autre que l'inégalité (4.15). En conséquence, dans le reste de cette section, on ne considère pas ces inégalités. Ces dernières seront traitées en même temps que les contraintes de capacité (4.15). Dans le théorème suivant, on considère les contraintes triviales (4.17).

Théorème 5.7 *Les inégalités* (4.17) *définissent des facettes pour $P_U(D, v_0, K)$.*

Preuve. Soit $a^* = (u^*, v^*)$ un arc quelconque de A. On montre que la contrainte triviale $y_{a^*} \leq 1$ définit une facette du polytope du 1-PRLPA unitaire. Pour cela, notons $F^t_{a^*}$ la face induite par l'inégalité (4.17) correspondant à l'arc a^*, c'est-à-dire,

$$F^t_{a^*} = \{(x, y) \in P_U(D, v_0, K) : y_{a^*} = 1\}.$$

Notons par $b^T x + c^T y \leq \alpha$ l'inégalité (4.17) correspondant à l'arc a^*. Soit $r^T x + s^T y \leq \beta$ une inégalité valide qui définit une facette F de $P_U(D, v_0, K)$. Supposons que $F^t_{a^*} \subseteq F$. On montre qu'il existe $\rho \in \mathbb{R}$, $\lambda = (\lambda^k_v, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$. (Nous rappelons que M_{CF} est la matrice des contraintes de conservation de flots.)

Par la proposition 5.2, il existe une solution (L, T_1) avec $T_1 = \delta(u^*)$. Comme l'arc a^* appartient à T_1, $(x^L, y^{T_1}) \in F^t_{a^*}$. Considérons deux sommets u et v différents de u^*. On peut ajouter à T_1 les deux arcs (u, v) et (v, u) pour obtenir une nouvelle solution (L, T_2) avec $T_2 = T_1 \cup \{(u, v), (v, u)\}$. Comme (x^L, y^{T_2}) appartient également à $F^t_{a^*}$, on a $rx^L + sy^{T_1} = rx^L + sy^{T_2}$. Ceci implique que $s_{(u,v)} + s_{(v,u)} = 0$. Ce résultat reste vrai si u et v sont deux sommets quelconques différents de v^*. On déduit alors que

$$s_{(u,v)} + s_{(v,u)} = 0 \text{ pour tout } u \neq v \in V,\ a^* \neq (u, v), (v, u). \tag{5.8}$$

Considérons trois sommets distincts u, v et w de V tels que $a^* \notin \{(u, v), (v, w), (w, u)\}$. Soit $T = A \setminus \{(u, v), (v, w), (w, u)\}$. Remarquons d'abord que T induit un graphe Eulérien. De plus, comme $n \geq 5$, le graphe orienté induit par T est $(p + 1)$-arc connexe. La proposition 5.6 implique qu'il existe L' tel que les solutions (L', T) et (L', A) sont réalisables pour le 1-PLRPA unitaire. Comme $a^* \in T$, ces solutions appartiennent à $F^t_{a^*}$, ce qui implique que

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0, \qquad \begin{array}{l} \forall\ u \neq v \neq w \neq u \in V, \\ a^* \neq (u, v), (v, w), (w, u). \end{array} \tag{5.9}$$

Soit $\bar{v} \in V$ un sommet différent de u^*, v^*. Posons $\mu_{\bar{v}} = 0$ et $\mu_v = -s_{(\bar{v},v)}$ pour tout $v \in V \setminus \{\bar{v}\}$. On a alors $s_{(\bar{v},v)} = \mu_{\bar{v}} - \mu_v$ pour tout $v \in V \setminus \{\bar{v}\}$. Par (5.8), on déduit que $s_{(v,\bar{v})} = \mu_v - \mu_{\bar{v}}$ pour tout $v \in V \setminus \{\bar{v}\}$. Considérons deux sommets quelconques $u, v \in V \setminus \{\bar{v}\}$ tels que $a^* \neq (u, v)$. Par (5.9), on a $s_{(u,v)} + s_{(v,\bar{v})} + s_{(\bar{v},u)} = 0$. On déduit alors que

$$s_{(u,v)} = \mu_u - \mu_v, \qquad \forall\ (u, v) \in A \setminus \{a^*\}. \tag{5.10}$$

Posons maintenant $\rho = \mu_{v^*} - \mu_{u^*} + s_{(u^*,v^*)}$. On a alors $s_{(u^*,v^*)} = \mu_{u^*} - \mu_{v^*} + \rho$. Le vecteur s n'est rien d'autre qu'une combinaison linéaire des contraintes de conservation (4.11) et de la contrainte $y_{a^*} = 1$.

Considérons maintenant le vecteur r^k associé à une demande de K, disons $\bar{k}$. Soit (u, v) un arc de $A^{\bar{k}} \backslash \delta^{\text{out}}(o^{\bar{k}})$. Considérons l'ensemble $B = \{(o^{\bar{k}}, u), (u, v), (v, d^{\bar{k}}), (o^{\bar{k}}, v)\}$. Le graphe induit par $A \setminus B$ est p-arc connexe. La proposition 5.5 implique qu'il existe deux solutions, disons (L^1, A) et (L^2, A) telles que $L^1_{\bar{k}} = \{(o^{\bar{k}}, u), (u, v), (v, d^{\bar{k}})\}$, $L^2_{\bar{k}} = \{(o^{\bar{k}}, v), (v, d^{\bar{k}})\}$ et $L^2_k = L^3_k$ pour toute demande $k \neq \bar{k}$. Il est clair que les vecteurs d'incidence de ces deux solutions appartiennent à $F^t_{a^*}$. On a alors $x^{L^1} + y^A = x^{L^2} + y^A$, ce qui implique que $r^{\bar{k}}_{(o^{\bar{k}},u)} + r^{\bar{k}}_{(u,v)} = r^{\bar{k}}_{(o^{\bar{k}},v)}$. Comme $\bar{k}$ est une demande quelconque de K et (u, v) un arc quelconque de $A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$, on peut généraliser le résultat comme suit

$$r^k_{(o^k,u)} + r^k_{(u,v)} = r^k_{(o^k,v)} \qquad \forall\, k \in K, \forall\, (u, v) \in A^k \setminus \delta^{\text{out}}(o^k). \tag{5.11}$$

Soit k une demande de K. Posons $\lambda^k_{o^k} = 0$ et $\lambda^k_v = -r^k_{(o^k,v)}$ pour tout $v \in V \setminus \{o^k\}$. Alors $r^k_{(o^k,v)} = \lambda^k_{o^k} - \lambda^k_v$ pour tout $v \in V \setminus \{o^k\}$. Considérons un arc (u, v) de $A^k \setminus \delta(o^k)$. D'après l'équation (5.11) associée à l'arc (u, v) et d'après les valeurs de r^k associées aux arcs sortant de o^k, on déduit que $r^k_{(u,v)} = \lambda^k_u - \lambda^k_v$. Comme k est une demande quelconque de K et (u, v) un arc quelconque de A^k, il s'ensuit que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\, k \in K, \forall\, (u, v) \in A^k. \tag{5.12}$$

Par conséquent, le vecteur r n'est rien d'autre qu'une combinaison linéaire des équations (4.12). On a alors $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$, ce qui termine la preuve. □

Nous étudions maintenant la structure faciale des contraintes triviales associées à x.

Nous remarquons tout d'abord que les inégalités $x^k_a \leq 1$ pour tout arc $a \in A$ pour toute demande $k \in K^a$ sont dominées par les contraintes de capacité (4.15). En conséquence, nous étudions uniquement dans cette partie les inégalités triviales (4.18).

Soient k^* une demande de K et $a^* = (u^*, v^*)$ un arc de A^{k^*}. Soit $F^{k^*}_{a^*}$ la face induite par l'inégalité triviale (4.18) correspondant à l'arc a^* et la demande k^*, c'est-à-dire,

$$F^{k^*}_{a^*} = \{(x, y) \in P_U(D, v_0, K) : x^{k^*}_{a^*} = 0\}.$$

Théorème 5.8 *Les inégalités* (4.18) *définissent des facettes pour* $P_U(D, v_0, K)$.

Preuve. Soit $b^T x + c^T y \geq \alpha$ l'inégalité (4.18) correspondant à l'arc a^* et la demande k^*. Soit $r^T x + s^T y \geq \beta$ une inégalité valide définissant une facette F de $P_U(D, v_0, K)$. Supposons que $F_{a^*}^{k^*} \subseteq F$. Nous montrons qu'il existe, $\rho \in \mathbb{R}$, $\lambda = (\lambda_v^k, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = \rho(b, c) + (\lambda, \mu) M_{CF}$.

Soit $\bar{v}$ un sommet de $V \setminus \{u^*, v^*\}$. La proposition 5.2 assure qu'il existe une solution, disons (L^1, T^1), avec $T^1 = \delta(\bar{v})$. Comme a^* n'appartient à aucun chemin de L^1, alors (x^{L^1}, y^{T^1}) apppartient à $F_{a^*}^{k^*}$. Soient u et v deux sommets de $V \setminus \{\bar{v}\}$. La solution (L^1, T^2), avec $T^2 = T^1 \cup \{(u, v), (v, u)\}$, appartient également à $F_{a^*}^{k^*}$. On a alors $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^2}$, ce qui implique que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V \setminus \{\bar{v}\}. \tag{5.13}$$

Considérons maintenant trois sommets distincts u, v et w de $V \setminus \{\bar{v}\}$. La solution (L^1, T^3), avec $T^3 = T^1 \cup \{(u, v), (v, w), (w, u)\}$, appartient aussi à $F_{a^*}^{k^*}$. On a donc $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^3}$, ce qui permet de déduire que

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall\ u \neq v \neq w \neq u \in V \setminus \{\bar{v}\}. \tag{5.14}$$

Comme $n \geq 5$, il existe deux sommets distincts appartenant à $V \setminus \{u^*, v^*, \bar{v}\}$, disons $\tilde{v}$ et $\hat{v}$. En réitérant le même raisonnement en considérant successivement les sommets $\tilde{v}$ et $\hat{v}$ à la place du sommet $\bar{v}$, on obtient, en combinant avec les équations (5.13) et (5.14)

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V, \tag{5.15}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \begin{array}{l} \forall\ u \neq v \neq w \neq u \in V, \\ \{u, v, w\} \neq \{\bar{v}, \tilde{v}, \hat{v}\}. \end{array} \tag{5.16}$$

Posons $\mu_{\bar{v}} = 0$ et $\mu_v = -s_{(\bar{v},v)}$ pour tout $v \in V \setminus \{\bar{v}\}$. On a alors $s_{(\bar{v},v)} = \mu_{\bar{v}} - \mu_v$ pour tout $v \in V \setminus \{\bar{v}\}$. Les équations (5.15) impliquent que $s_{(v,\bar{v})} = \mu_v - \mu_{\bar{v}}$ pour tout $v \in V \setminus \{\bar{v}\}$. Soient u et v deux sommets distincts de $V \setminus \{\bar{v}\}$ tels que $(u, v) \notin \{(\tilde{v}, \hat{v}), (\hat{v}, \tilde{v})\}$. Par les équations (5.16), on obtient que $s_{(\bar{v},u)} + s_{(u,v)} + s_{(v,\bar{v})} = 0$. Comme u et v sont deux sommets quelconques tels que $(u, v) \notin \{(\tilde{v}, \hat{v}), (\hat{v}, \tilde{v})\}$ et d'après les valeurs de s associées aux arcs incidents à $\bar{v}$, on obtient

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u, v) \in A \setminus \{(\tilde{v}, \hat{v}), (\hat{v}, \tilde{v})\}. \tag{5.17}$$

Considérons l'équation (5.16) associée aux sommets $\tilde{v}, \hat{v}, u^*$. On a alors $s_{(\tilde{v},\hat{v})} + s_{(\hat{v},u^*)} + s_{(u^*,\tilde{v})} = 0$, ce qui implique que $s_{(\tilde{v},\hat{v})} = \mu_{\tilde{v}} - \mu_{\hat{v}}$. De plus, l'équation (5.15) associée à $\tilde{v}$ et $\hat{v}$ implique que $s_{(\hat{v},\tilde{v})} = \mu_{\hat{v}} - \mu_{\tilde{v}}$. Il s'ensuit alors que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u, v) \in A. \tag{5.18}$$

Le vecteur s n'est rien d'autre qu'une combinaison linéaire des contraintes de conservation de flot associées à y.

Considérons maintenant les valeurs de r associées aux demandes de $K \setminus \{k^*\}$. Soit $\bar{k}$ une demande de $K \setminus \{k^*\}$ et (u, v) un arc quelconque de $A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$. On suppose par la suite que v est différent de $d^{\bar{k}}$. La preuve du cas $v = d^{\bar{k}}$ étant similaire, elle n'est donc pas donnée.
On définit maintenant un ensemble d'arcs, disons B^1. Pour cela, il est nécessaire de considérer selon certains cas un sommet de $V \setminus \{o^{k^*}, d^{k^*}\}$, disons w, défini de la manière suivante. Supposons que $(o^{k^*}, d^{k^*}) = (u, v)$. Si $a^* \notin \{(u, d^{\bar{k}}), (d^{\bar{k}}, v)\}$, alors $w = d^{\bar{k}}$. Autrement, $w \in V \setminus \{o^{\bar{k}}, u, v, d^{\bar{k}}\}$. Supposons maintenant que $(o^{k^*}, d^{k^*}) = (u^*, v^*)$. Deux cas sont à considérer. Supposons que $v = o^{k^*}$. Si $u \neq d^{k^*}$, alors $w = u$. Sinon, $w \in V \setminus \{o^{\bar{k}}, u, v, d^{\bar{k}}\}$. Supposons maintenant que $v \neq o^{k^*}$. Si $d^{\bar{k}}$ (respectivement u) n'appartient pas à $\{o^{k^*}, d^{k^*}\}$, alors $w = d^{\bar{k}}$ (respectivement $w = u$). Autrement, $\{u, d^{\bar{k}}\} = \{o^{k^*}, d^{k^*}\}$ et w appartient alors à $V \setminus \{o^{\bar{k}}, u, v, d^{\bar{k}}\}$. On définit l'ensemble d'arcs

$$B^1 = \begin{cases} \{(o^{k^*}, d^{k^*})\} & \text{si } (o^{k^*}, d^{k^*}) \notin \{(u, v), (u^*, v^*)\}, \\ \{(o^{k^*}, w), (w, d^{k^*})\} & \text{sinon.} \end{cases}$$

Soit B^2 l'ensemble d'arcs $\{(o^{\bar{k}}, u), (o^{\bar{k}}, v), (u, v), (v, d^{\bar{k}})\}$. Il est facile de voir, par énumération, que B^1 et B^2 sont arc-disjoints. Par ailleurs, le graphe induit par $A \setminus (B^1 \cup B^2)$ est $(n-4)$-arc connexe, ce qui implique qu'il est $(p-1)$-arc connexe, puisque $n \geq 2p+1$ et $p \geq 2$. D'après le corollaire du théorème des branchements disjoints d'Edmonds [90] (Corollaire 53.1a, p. 905), il existe $p-1$ arborescences disjointes $\Gamma_0, \Gamma_1, \ldots, \Gamma_{p-2}$ dans le graphe induit par $A \setminus (B^1 \cup B^2)$. On associe les $p-2$ dernières arborescences aux demandes de $K \setminus \{\bar{k}, k^*\}$. Soit Γ_k l'arborescence associée à la demande k. On suppose, sans perte de généralité, que Γ_0 (respectivement Γ_k, $k \in K \setminus \{\bar{k}, k^*\}$) a pour racine v_0 (respectivement o^k). Pour chaque demande $k \in K \setminus \{\bar{k}, k^*\}$, il existe dans Γ_k un chemin élémentaire de o^k à d^k que l'on note L^2_k. On pose $L^2_{k^*} = B^1$ et $L^2_{\bar{k}} = B^2 \setminus \{(o^{\bar{k}}, v)\}$. On a alors un ensemble $L^2 = \{L^2_1, L^2_2, \ldots, L^2_p\}$ de $o^k d^k$-chemins élémentaires arc-disjoints puisque les arborescences sont disjointes et n'utilisent aucun arc de $B^1 \cup B^2$. On a également une v_0-arborescence Γ_0 qui n'intersecte aucun arc de L^2. D'après la proposition 5.1, (L^2, A) est donc réalisable pour le 1-PRLPA unitaire. Par ailleurs, il est clair que l'arc (u^*, v^*) n'appartient pas à L_{k^*}, ce qui implique que le vecteur d'incidence de (L^2, A) est un point de $F^{k^*}_{a^*}$.
De la même manière, on peut construire une solution (L^3, A) telle que $L^3_k = L^2_k$ pour tout $k \neq \bar{k}$ et $L^3_{\bar{k}} = B^2 \setminus \{(o^{\bar{k}}, u), (u, v)\}$. Comme (L^3, A) appartient également à $F^{k^*}_{a^*}$, on a alors $x^{L^2} + y^A = x^{L^3} + y^A$, ce qui implique que

$$r^{\bar{k}}_{(o^{\bar{k}}, u)} + r^{\bar{k}}_{(u,v)} = r^{\bar{k}}_{(o^{\bar{k}}, v)} \qquad \forall\ (u, v) \in A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}}). \tag{5.19}$$

Posons $\lambda^{\bar{k}}_{o^{\bar{k}}} = 0$ et $\lambda^{\bar{k}}_v = -r^{\bar{k}}_{(o^{\bar{k}},v)}$ pour tout $v \in V \setminus \{o^{\bar{k}}\}$. On a alors $r^{\bar{k}}_{(o^{\bar{k}},v)} = \lambda^{\bar{k}}_{o^{\bar{k}}} - \lambda^{\bar{k}}_v$ pour tout $v \in V \setminus \{o^{\bar{k}}\}$. Considérons un arc $(u,v) \in A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$. D'après l'équation (5.19) et les valeurs de $r^{\bar{k}}$ associées aux arcs sortant de $o^{\bar{k}}$, on déduit que $r^{\bar{k}}_{(u,v)} = \lambda^{\bar{k}}_u - \lambda^{\bar{k}}_v$ pour tout $(u,v) \in A^{\bar{k}} \setminus \delta^{\text{out}}(o^{\bar{k}})$. Comme $\bar{k}$ est une demande quelconque de $k \setminus \{k^*\}$, il s'ensuit que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v, \qquad \forall\, k \in K \setminus \{k^*\}, \forall\, (u,v) \in A^k. \tag{5.20}$$

Considérons maintenant la demande k^*. Supposons d'abord que $a^* = (o^{k^*}, d^{k^*})$. Soient u et v deux sommets distincts de $V \setminus \{o^{k^*}, d^{k^*}\}$. Soit (L^4, A) la solution telle que

$$L^4_k = \begin{cases} \{(o^{k^*},u),(u,v),(v,d^{k^*})\} & \text{si } k = k^*, \\ \{(o^k,v_0),(v_0,d^k)\} & \text{si } k \neq k^* \text{ et } v_0 \notin \{u,v\}, \\ \{(o^k,d^k)\} & \text{sinon.} \end{cases}$$

Les chemins L^4_k, $k \in K$ sont arc-disjoints. De plus, il est clair qu'il existe une v_0-arborescence Γ dans $T \setminus L$ couvrant les origines o^k, $k \in K$. D'après la proposition 5.1, la solution (L^4, A) est réalisable pour le 1-PRLPA unitaire. De plus, a^* n'appartient pas à $L^4_{k^*}$, ce qui implique que (x^{L^4}, y^A) appartient à $F^{k^*}_{a^*}$. Considérons maintenant la solution (L^5, A) avec $L^5_{k^*} = \{(o^{k^*},v),(v,d^{k^*})\}$ et $L^5_k = L^4_k$ pour toute demande k différente de k^*. Il est clair que (x^{L^5}, y^A) appartient également à $F^{k^*}_{a^*}$. Comme u et v sont deux sommets distincts différents de o^{k^*} et d^{k^*}, ceci implique que

$$r^{k^*}_{(o^{k^*},u)} + r^{k^*}_{(u,v)} = r^{k^*}_{(o^{k^*},v)} \qquad \forall\, u \neq v \in V \setminus \{o^{k^*}, d^{k^*}\}. \tag{5.21}$$

Notons par B^2 l'ensemble d'arcs $\{(o^{k^*},u),(u,d^{k^*}),(o^{k^*},v),\ (v,d^{k^*})\}$. Le graphe induit par $A \setminus B^2$ est $(n-3)$-arc connexe et, par conséquent, p-arc connexe puisque $n \geq 2p+1$ et $p \geq 2$. La proposition 5.5 assure alors qu'il existe deux solutions réalisables, disons (L^6, A) et (L^7, A), telles que $L^6_{k^*} = \{(o^{k^*},v),(v,d^{k^*})\}$, $L^7_{k^*} = \{(o^{k^*},w),(w,d^{k^*})\}$ et $L^6_k = L^7_k$ pour tout $k \neq k^*$. Comme a^* n'appartient pas à $L^6_{k^*}$ et $L^7_{k^*}$, ces deux solutions appartiennent à $F^{a^*}_{k^*}$. On déduit alors que

$$r^{k^*}_{(o^{k^*},u)} + r^{k^*}_{(u,d^{k^*})} = r^{k^*}_{(o^{k^*},v)} + r^{k^*}_{(v,d^{k^*})} \qquad \forall\, u \neq v \in V \setminus \{o^{k^*}, d^{k^*}\}. \tag{5.22}$$

Posons $\lambda^{k^*}_{o^{k^*}} = 0$ et $\lambda^{k^*}_v = -r^{k^*}_{(o^{k^*},v)}$ pour tout $v \in V \setminus \{o^{k^*}, d^{k^*}\}$. On a donc $r^{k^*}_{(o^{k^*},v)} = \lambda^{k^*}_{o^{k^*}} - \lambda^{k^*}_v$ pour tout $v \in V \setminus \{o^{k^*}, d^{k^*}\}$. Considérons un arc (u,v) de $A^{k^*} \setminus (\delta(o^{k^*}) \cup \delta(d^{k^*}))$. Les équations (5.21) assurent que l'on a

$$r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v \qquad \forall\, (u,v) \in A^{k^*} \setminus (\delta(o^{k^*}) \cup \delta(d^{k^*})). \tag{5.23}$$

Considérons un sommet $\hat{v}$ de $V \setminus \{o^{k^*}, d^{k^*}\}$ et posons $\lambda^{k^*}_{d^{k^*}} = -r^{k^*}_{(o^{k^*},\hat{v})} - r^{k^*}_{(\hat{v},d^{k^*})}$. On obtient alors $r^{k^*}_{(\hat{v},d^{k^*})} = \lambda^{k^*}_{\hat{v}} - \lambda^{k^*}_{d^{k^*}}$. Soit v un sommet quelconque de $V \setminus \{o^{k^*}, d^{k^*}, \hat{v}\}$. Les

équations (5.22) impliquent que $r^{k^*}_{(o^{k^*},\hat{v})}+r^{k^*}_{(\hat{v},d^{k^*})} = r^{k^*}_{(o^{k^*},v)}+r^{k^*}_{(v,d^{k^*})}$. D'après les valeurs de r^{k^*} associées aux arcs sortant de o^{k^*} et à l'arc $(\hat{v}, d^{k^*})$, il s'ensuit que $r^{k^*}_{(v,d^{k^*})} = \lambda^{k^*}_v - \lambda^{k^*}_{d^{k^*}}$. Comme v est quelconque et d'après (5.23), on obtient

$$r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v \qquad \forall\ (u,v) \in A^{k^*} \setminus \{a^*\}. \tag{5.24}$$

Posons $\rho = r^{k^*}_{(o^{k^*},d^{k^*})} - r^{k^*}_{(o^{k^*},\hat{v})} - r^{k^*}_{(\hat{v},d^{k^*})}$. On obtient alors $r^{k^*}_{(o^{k^*},d^{k^*})} = \lambda^{k^*}_{o^{k^*}} - \lambda^{k^*}_{d^{k^*}} + \rho$. On déduit alors, d'après (5.24), que r^{k^*} est une combinaison linéaire des contraintes de conservation de flot associées à la demande k^* et de l'équation $x^{k^*}_{a^*} = 0$.

Considérons maintenant le cas où a^* n'est pas un arc sortant de o^{k^*}. (La preuve du cas où a^* est un arc sortant de o^{k^*} différent de (o^{k^*}, d^{k^*}) étant similaire à la preuve de ce cas, elle ne sera pas donnée.) Soient u, v deux sommets distincts de $V \setminus \{o^{k^*}\}$ avec $(u,v) \in A^{k^*}$ et $(u,v) \neq a^*$. On considère l'ensemble d'arcs B^3 défini par

$$B^3 = \begin{cases} \{(o^{k^*},u),(u,v),(v,d^{k^*}),(o^{k^*},v)\} & \text{si } a^* \neq (v,d^{k^*}), \\ \{(o^{k^*},u),(u,v),(v,w),(w,d^{k^*}),(o^{k^*},v)\} & \text{sinon.} \end{cases}$$

Le graphe induit par l'ensemble d'arcs $A \setminus B^3$ est p-arc connexe. D'après la proposition 5.5, il existe une solution, disons (L^8, A) avec

$$L^8_{k^*} = \begin{cases} \{(o^{k^*},u),(u,v),(v,d^{k^*})\} & \text{si } a^* \neq (v,d^{k^*}), \\ \{(o^{k^*},u),(u,v),(v,w),(w,d^{k^*})\} & \text{sinon.} \end{cases}$$

Comme a^* n'appartient pas à $L^8_{k^*}$, alors le vecteur d'incidence de (L^8, A) est un point de $F^{k^*}_{a^*}$. Par ailleurs, il existe une autre solution, disons (L^9, A), appartenant aussi à $F^{k^*}_{a^*}$, telle que

$$L^9_k = \begin{cases} \{(o^{k^*},v),(v,d^{k^*})\} & \text{si } k = k^* \text{ et } a^* \neq (v,d^{k^*}), \\ \{(o^{k^*},v),(v,w),(w,d^{k^*})\} & \text{si } k = k^* \text{ et } a^* = (v,d^{k^*}), \\ L^8_k & \text{sinon.} \end{cases}$$

Ceci implique alors que

$$r^{k^*}_{(o^{k^*},u)} + r^{k^*}_{(u,v)} = r^{k^*}_{(o^{k^*},v)} \qquad \forall\ u \neq v \in V \setminus \{o^{k^*}\}, (u,v) \neq a^*. \tag{5.25}$$

Posons $\lambda^{k^*}_{o^{k^*}} = 0$ et $\lambda^{k^*}_v = -r^{k^*}_{(o^{k^*},v)}$ pour tout $v \in V \setminus \{o^{k^*}\}$. On a donc $r^{k^*}_{(o^{k^*},v)} = \lambda^{k^*}_{o^{k^*}} - \lambda^{k^*}_v$ pour tout $v \in V \setminus \{o^{k^*}\}$. Considérons un arc (u,v) de $A^{k^*} \setminus \delta(o^{k^*})$. Les équations (5.25) impliquent que l'on a

$$r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v \qquad \forall\ (u,v) \in A^{k^*} \setminus \{a^*\}. \tag{5.26}$$

En posant $\rho = r^{k^*}_{a^*} + r^{k^*}_{(o^{k^*},u^*)} - r^{k^*}_{(o^{k^*},v^*)}$, on obtient alors que $r^{k^*}_{a^*} = \lambda^{k^*}_{u^*} - \lambda^{k^*}_{v^*} + \rho$, ce qui implique que r^{k^*} est une combinaison linéaire des contraintes de conservation de flot associées à k^* et de la contrainte $x^{k^*}_{a^*} = 0$. D'après les équations (5.18) et (5.20), il s'ensuit que $(r,s) = \rho(b,c) + (\lambda,\mu)M_{CF}$, ce qui termine la preuve. □

Nous considérons maintenant les propriétés faciales des contraintes de capacité (4.15).

5.1.2.2 Contraintes de capacité

Soient $a^* = (u^*, v^*)$ un arc de A et F_{a^*} la face induite par la contrainte de capacité (4.15) correspondant à l'arc a^*, c'est-à-dire,

$$F_{a^*} = \{(x, y) \in P_U(D, v_0, K) : y_{a^*} - x_{a^*}(K^{a^*}) = 0\}.$$

Théorème 5.9 *Les inégalités de capacité* (4.15) *définissent des facettes pour* $P_U(D, v_0, K)$.

Preuve. Soit $b^T x + c^T y \geq \alpha$ l'inégalité (4.15) correspondant à l'arc a^*. Soit $r^T x + s^T y \geq \beta$ une inégalité valide définissant une facette F de $P_U(D, v_0, K)$. Supposons que $F_{a^*} \subseteq F$. On montre qu'il existe ρ, $\lambda = (\lambda_v^k, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$.

Soit $\bar{v}$ un sommet de $V \setminus \{u^*, v^*\}$. La proposition 5.2 assure qu'il existe une solution, disons (L^1, T^1), avec $T^1 = \delta(\bar{v})$. Comme a^* n'appartient pas à T^1, alors (x^{L^1}, y^{T^1}) appartient à F_{a^*}. Soient u et v deux sommets de $V \setminus \{\bar{v}\}$ tels que $(u^*, v^*) \notin \{(u, v), (v, u)\}$. La solution (L^1, T^2), avec $T^2 = T^1 \cup \{(u, v), (v, u)\}$, appartient également à F_{a^*}. On a alors $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^2}$, ce qui implique que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V \setminus \{\bar{v}\}, (u^*, v^*) \notin \{(u, v), (v, u)\}. \tag{5.27}$$

Considérons maintenant trois sommets distincts u, v et w de $V \setminus \{\bar{v}\}$ tels que $(u^*, v^*) \notin \{(u, v), (v, w), (w, u)\}$. La solution (L^1, T^3), avec $T^3 = T^1 \cup \{(u, v), (v, w), (w, u)\}$, appartient aussi à $F_{a^*}^{k^*}$. On a donc $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^3}$, ce qui nous permet de déduire que

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \begin{array}{l} \forall\ u \neq v \neq w \neq u \in V \setminus \{\bar{v}\}, \\ (u^*, v^*) \notin \{(u, v), (v, w), (w, u)\}. \end{array} \tag{5.28}$$

Comme $n \geq 5$, il existe deux sommets distincts appartenant à $V \setminus \{u^*, v^*, \bar{v}\}$, disons $\tilde{v}$ et $\hat{v}$. En réitérant le même raisonnement en considérant successivement les sommets $\tilde{v}$ et $\hat{v}$ à la place du sommet $\bar{v}$, on obtient, en combinant avec les équations (5.27) et (5.28)

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V, (u^*, v^*) \notin \{(u, v), (v, u)\} \tag{5.29}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \begin{array}{l} \forall\ u \neq v \neq w \neq u \in V, \\ \{u, v, w\} \neq \{\bar{v}, \tilde{v}, \hat{v}\}, \\ (u^*, v^*) \notin \{(u, v), (v, w), (w, u)\}. \end{array} \tag{5.30}$$

Posons $\mu_{\bar{v}} = 0$ et $\mu_v = -s_{(\bar{v},v)}$ pour tout $v \in V \setminus \{\bar{v}\}$. On a alors $s_{(\bar{v},v)} = \mu_{\bar{v}} - \mu_v$ pour tout $v \in V \setminus \{\bar{v}\}$. Les équations (5.29) impliquent que $s_{(v,\bar{v})} = \mu_v - \mu_{\bar{v}}$ pour tout $v \in V \setminus \{\bar{v}\}$. Soient u et v deux sommets distincts de $V \setminus \{\bar{v}\}$ tels que $(u,v) \notin \{(\tilde{v},\hat{v}),(\hat{v},\tilde{v}),(u^*,v^*)\}$. Par les équations (5.30), on obtient que $s_{(\bar{v},u)} + s_{(u,v)} + s_{(v,\bar{v})} = 0$. Comme u et v sont choisis arbitrairement et d'après les valeurs de s associées aux arcs incidents à $\bar{v}$, on obtient

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u,v) \in A \setminus \{a^*,(\tilde{v},\hat{v}),(\hat{v},\tilde{v})\}. \tag{5.31}$$

Considérons l'équation (5.30) associée aux sommets $u^*, \tilde{v}, \hat{v}$. On a $s_{(u^*,\tilde{v})} + s_{(\tilde{v},\hat{v})} + s_{(\hat{v},u^*)} = 0$, ce qui implique, d'après les valeurs de $s_{(u^*,\tilde{v})}$ et $s_{(\hat{v},u^*)}$, que $s_{(\tilde{v},\hat{v})} = \mu_{\tilde{v}} - \mu_{\hat{v}}$. De plus, l'équation (5.29) implique que $s_{(\hat{v},\tilde{v})} = \mu_{\hat{v}} - \mu_{\tilde{v}}$. Posons $\rho = s_{(u^*,v^*)} + s_{(\bar{v},u^*)} - s_{(\bar{v},v^*)}$. On obtient alors

$$s_{(u,v)} = \begin{cases} \mu_u - \mu_v + \rho & \text{si } (u,v) = a^*, \\ \mu_u - \mu_v & \text{sinon,} \end{cases} \qquad \forall\ (u,v) \in A. \tag{5.32}$$

On s'intéresse maintenant au vecteur r. Considérons une demande de K, disons la demande 1. Considérons dans un premier temps que $a^* \in \{(o^1,d^1),(d^1,o^1)\}$. Soient u, v deux sommets de $V \setminus \{o^1,d^1\}$. Soient $T^4 = A \setminus \{(u^*,v^*),(v^*,u^*)\}$ et $B^1 = \{(o^1,u),(u,v),(v,d^1),(o^1,v)\}$. Supposons d'abord que $n \geq 6$. Le graphe induit par $T^4 \setminus B^1$ est alors p-arc connexe, puisqu'il est $(n-4)$-arc connexe. Comme le graphe induit par T^4 est Eulérien, d'après la proposition 5.5, il existe deux solutions, disons (L^2,T^4) et (L^3,T^4) telles que $L^2_1 = \{(o^1,u),(u,v),(v,d^1)\}$, $L^3_1 = \{(o^1,v),(v,d^1)\}$ et $L^2_k = L^3_k$ pour tout $k \neq 1$.
Supposons maintenant que $n = 5$. Alors $K = \{1,2\}$. Notons w le sommet de $V \setminus \{o^1,d^1,u,v\}$. Comme chaque sommet est incident à au plus une demande, alors $\{o^2,d^2\} \subseteq \{u,v,w\}$. Notons B^2 l'ensemble des arcs du circuit passant successivement par v, u et w, *i.e.*, $B^2 = \{(v,u),(u,w),(w,v)\}$ et L^4_2 le chemin de o^2 à d^2 dans B^2. Comme $T^4 \setminus \{B^1 \cup B^2\}$ induit un graphe fortement connexe, il existe une v_0-arborescence couvrant les origines de K utilisant uniquement les arcs de $T^4 \setminus \{B^1 \cup B^2\}$. D'après la proposition 5.1, il s'ensuit que la solution (L^4,T^4) avec $L^4_1 = \{(o^1,u),(u,v),(v,d^1)\}$ est réalisable. De plus, la solution (L^5,T^4) avec $L^5_1 = \{(o^1,v),(v,d^1)\}$ et $L^5_2 = L^4_2$ est également réalisable. Comme a^* n'appartient pas à T^4, les vecteurs d'incidence des solutions (L^2,T^4), (L^3,T^4), (L^4,T^4) et (L^5,T^4) appartiennent à F_{a^*}. Comme u et v sont deux sommets quelconques de $V \setminus \{o^1,d^1\}$, on déduit alors que

$$r^1_{(o^1,u)} + r^1_{(u,v)} = r^1_{(o^1,v)} \qquad \forall\ u \neq v \in V \setminus \{o^1,d^1\}. \tag{5.33}$$

Considérons maintenant l'ensemble $B^3 = \{(o^1,u),(u,d^1),(o^1,v),(v,d^1)\}$. Le graphe induit par $T^4 \setminus B^3$ est p-arc connexe si $n \geq 6$. Si $n = 5$, B^3 n'intersecte aucun arc

de B^2 et le graphe induit par $T^4 \setminus \{B^2 \cup B^3\}$ est fortement connexe. En utilisant le même raisonnement que celui utilisé pour l'ensemble B^1, on montre alors qu'il existe deux solutions, disons (L^6, T^4) et (L^7, T^4), telles que $L_1^6 = \{(o^1, u), (u, d^1)\}$, $L_1^7 = \{(o^1, v), (v, d^1)\}$ et $L_k^6 = L_k^7$ pour toute demande $k \neq 1$. On obtient alors

$$r^1_{(o^1,u)} + r^1_{(u,d^1)} = r^1_{(o^1,v)} + r^1_{(v,d^1)} \qquad \forall\ u \neq v \in V \setminus \{o^1, d^1\}. \tag{5.34}$$

Posons $\lambda^1_{o^1} = 0$ et $\lambda^1_v = -r^1_{(o^1,v)}$ pour tout $v \in V \setminus \{o^1, d^1\}$. On a $r^1_{(o^1,v)} = \lambda^1_{o^1} - \lambda^1_v$ pour tout $v \in V \setminus \{o^1, d^1\}$. Soit (u, v) un arc de A^1, avec u et v deux sommets de $V \setminus \{o^1, d^1\}$. Les équations (5.33) impliquent alors que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in A^1, u, v \in V \setminus \{o^1, d^1\}. \tag{5.35}$$

Soit $\bar{v}$ un sommet de $V \setminus \{o^1, d^1\}$. Posons $\lambda^1_{d^1} = -r^1_{(o^1,\bar{v})} - r^1_{(\bar{v},d^1)}$. On déduit que $r^1_{(\bar{v},d^1)} = \lambda^1_{\bar{v}} - \lambda^1_{d^1}$. Soit v un sommet de $V \setminus \{o^1, d^1, \bar{v}\}$. Les équations (5.34) impliquent alors que $r^1_{(o^1,v)} + r^1_{(v,d^1)} = r^1_{(o^1,\bar{v})} + r^1_{(\bar{v},d^1)}$. D'après les valeurs de r^1 associées aux arcs sortant de o^1 et à l'arc $(\bar{v}, d^1)$, on obtient $r^1_{(v,d^1)} = \lambda^1_v - \lambda^1_{d^1}$. Comme v est quelconque et d'après les équations (5.35), il s'ensuit que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in A^1 \setminus \{a^*\}. \tag{5.36}$$

En réitérant pour toute demande de K telle que $a^* \in \{(o^k, d^k), (d^k, o^k)\}$, on obtient alors

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\ k \in K, a^* \in \{(o^k, d^k), (d^k, o^k)\}, \forall\ (u, v) \in A^k \setminus \{a^*\}. \tag{5.37}$$

Considérons maintenant le cas où la demande 1 est telle que $(o^1, d^1) \neq a^* \neq (d^1, o^1)$. On prouve uniquement le cas lorsque $a^* \notin \delta(o^1)$. Le cas $a^* \notin \delta(d^1)$ pouvant être montré de la même manière, la preuve ne sera pas donnée. Soit (u, v) un arc de $A^1 \setminus \delta(o^1)$ tel que $a^* \neq (u, v)$. On suppose, sans perte de généralité, que $v \neq d^1$. Le cas $v = d^1$ étant similaire, la preuve de ce dernier n'est pas donnée. On montre alors que

$$r^1_{(o^1,u)} + r^1_{(u,v)} = r^1_{(o^1,v)} \qquad \forall\ (u, v) \in A^1 \setminus \delta(o^1), a^* \neq (u, v). \tag{5.38}$$

Supposons tout d'abord que $a^* = (v, d^1)$. Dans ce cas, le graphe induit par $A \setminus B^1$ est p-arc connexe. Comme D est Eulérien, par la proposition 5.5, les solutions (L^2, A) et (L^3, A) sont réalisables pour le 1-PRLPA unitaire. Comme dans ces deux solutions, la demande 1 est transportée sur l'arc a^*, la contrainte de capacité associée à a^* est satisfaite à l'égalité par les vecteurs d'incidence de ces deux solutions. Ces derniers

appartiennent donc à F_{a^*}. L'équation (5.38) est donc vérifiée. On suppose maintenant que a^* est différent de (v, d^1).

Considérons que $n \geq 6$. Si $a^* \neq (v, u)$, alors les solutions (L^2, T^4) et (L^3, T^4) appartiennent à F_{a^*} puisque a^* n'appartient pas à T^4 et l'équation (5.38) est vérifiée. Supposons maintenant que $a^* = (v, u)$. Posons $T^5 = A \backslash \{(v, u), (u, d^1), (d^1, v)\}$. On a $B^1 \subseteq T^5$. Par ailleurs, le graphe induit par $T^5 \backslash B^1$ est $(n-4)$-arc connexe, ce qui implique qu'il est p-arc connexe. Comme graphe induit par T^5 est Eulérien, d'après la proposition 5.5, il existe deux solutions, disons (L^8, T^5) et (L^9, T^5), telles que $L_1^8 = \{(o^1, u), (u, v), (v, d^1)\}$, $L_1^9 = \{(o^1, v), (v, d^1)\}$ et $L_k^8 = L_k^9$ pour tout $k \neq 1$. Comme les vecteurs d'incidence de ces deux solutions appartiennent à F_{a^*}, alors l'équation (5.38) est satisfaite.

Considérons maintenant que $n = 5$. Dans ce cas, $K = \{1, 2\}$. Supposons d'abord que a^* possède ses deux extrémités dans $\{o^1, u, v, d^1\}$. Il existe un sommet, disons w, appartenant à $V \setminus \{o^1, d^1, u, v, u^*, v^*\}$. Considérons l'ensemble T^6 correspondant à l'ensemble d'arcs A privé des arcs du circuit passant successivement par v, u et d^1, et du circuit passant successivement par les sommets o^1, d^1 et u *i.e.*, $T^6 = A \setminus \{(v, u), (u, d_1), (d_1, v), (o^1, d^1), (d^1, u), (u, o^1)\}$. Le graphe induit par T^6 est Eulérien. De plus, a^* n'appartient pas à T^6. Considérons L^{10} tel que $L_1^{10} = \{(o^1, u), (u, v), (v, d^1)\}$ et $L_2^{10} = \{(o^2, w), (w, d^2)\}$. (Si w est l'origine ou la destination de la demande 2, alors L_2^{10} correspond respectivement au chemin $\{(w, d^2)\}$ ou $\{(o^2, w)\}$.) L'ensemble d'arcs $\Gamma = \{(w, o^k) : k \in K\} \cup \{(v_0, w)\}$ définit une v_0-arborescence couvrant les origines o^1 et o^2 dans le graphe induit par T^6 et n'intersecte aucun arc de L^{10}. D'après la proposition 5.1, la solution (L^{10}, T^6) est réalisable. Comme $a^* \notin T^6$, cette solution appartient à F_{a^*}. De plus, il est clair que si $L_1^{10} = \{(o^1, v), (v, d^1)\}$, alors la solution (L^{10}, T^6) appartient toujours à F_{a^*}. (5.38) est donc vérifié.

Il reste maintenant à examiner le cas où $n = 5$ et a^* possède exactement une extrémité dans $\{o^1, u, v, d^1\}$. Notons par $\tilde{v}$ le sommet de $V \backslash \{o^1, u, v, d^1\}$. ($\tilde{v}$ est donc une extrémité de a^*). Considérons le graphe orienté $D' = (V', A')$ avec $A' = T^4 \setminus B^1$. Il est facile de voir par énumération que

$$|\delta^{\text{in}}_{D'-\tilde{v}}(W)| \begin{cases} = 1 & \text{si } W = \{v\} \text{ ou } W = \{u, v, d^1\}, \\ \geq 2 & \text{sinon.} \end{cases}$$

Considérons un sous-ensemble de sommets W de D'. Si $W = \{\tilde{v}\}$ ou $W = V \setminus \{\tilde{v}\}$, alors $|\delta^{\text{in}}_{D'}(W)| = 3$. Si $|\delta^{\text{in}}_{D'-\tilde{v}}(W)| \geq 2$, alors $|\delta^{\text{in}}_{D'}(W)| \geq 2$. On a alors

$$|\delta^{\text{in}}_{D'}(W)| \begin{cases} = 1 & \text{si } W = \{v\} \text{ et } a^* \in \{(\tilde{v}, v), (v, \tilde{v})\}, \\ = 1 & \text{si } W = \{u, v, \tilde{v}, d^1\} \text{ et } a^* \in \{(\tilde{v}, o^1), (o^1, \tilde{v})\}, \\ \geq 2 & \text{sinon.} \end{cases}$$

Si $a^* \notin \{(\tilde{v}, v), (v, \tilde{v})\}$ ou $a^* \in \{(\tilde{v}, v), (v, \tilde{v})\}$ et $v \cap \{v_0, o^2\} \neq \emptyset$, alors $|\delta^{\text{in}}_{D'}(W)| \geq |\{v_0, o^2\} \cap \overline{W}|$ pour tout $\emptyset \neq W \subset V$. En conséquence, d'après [90] [Theorem 53.1, p. 904], il existe deux arborescences disjointes couvrantes, disons Γ_1 et Γ_2, dans D' ayant respectivement v_0 et o^2 comme sommet racine. Soit L_2^{11} le chemin de o^2 à d^2 dans Γ_2. Comme Γ_1, L_1^2 et L_2^{11} sont arc-disjoints, et comme le graphe induit par T^4 contient L_1^2, par la proposition 5.1, (L^{11}, T^4), avec $L_1^{11} = L_1^2$, est une solution réalisable. Comme $a^* \notin T^4$, (L^{11}, T^4) appartient à F_{a^*}. De plus, la solution dans laquelle le chemin de la demande 1 est remplacé par le chemin L_1^3 appartient aussi à F_{a^*} et (5.38) est donc vérifié.

Supposons maintenant que $a^* \in \{(\tilde{v}, v), (v, \tilde{v})\}$, $v \neq v_0$ et $v \neq o^2$. Il est facile de voir que le graphe orienté $D' - v$ est 2-arc connexe. D'après [90] [Theorem 53.1, p. 904], ceci implique qu'il existe deux arborescences couvrantes disjointes, disons Γ'_1 et Γ'_2, dans $D' - v$ ayant respectivement v_0 et o^2 comme sommets d'origine. Soit $\hat{a}$ l'arc entrant de v dans D'. $\Gamma''_2 = \Gamma'_2 \cup \{\hat{a}\}$ est donc une o^2-arborescence de D' qui n'intersecte pas Γ'_1. De plus, comme $v_2 \neq o^1$ et $v_2 \neq o^2$, Γ'_1 couvre toutes les origines des demandes. En considérant le graphe induit par T^4 et les chemins L_1^{11} et L_2^{11} définis précédemment, nous avons alors une solution de F_{a^*}. De plus, la solution dans laquelle le chemin de la demande 1 est remplacé par le chemin L_1^3 appartient aussi à F_{a^*} et (5.38) est donc vérifié.

Posons $\lambda^1_{o^1} = 0$ et $\lambda^1_v = -r^1_{(o^1,v)}$ pour tout $v \in V \setminus \{o^1\}$. On a alors $r^1_{(o^1,v)} = \lambda^1_{o^1} - \lambda^1_v$ pour tout $v \in V \setminus \{o^1\}$. Considérons un arc (u, v) de $A^1 \setminus \delta(o^1)$ tel que $a^* \neq (u, v)$. Par (5.38), on obtient que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. En réitérant pour toute demande de K telle que $a^* \notin \{(o^k, d^k), (d^k, o^k)\}$, on obtient

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\, k \in K, a^* \notin \{(o^k, d^k), (d^k, o^k)\}, \forall\, (u, v) \in A^k \setminus \{a^*\}. \quad (5.39)$$

En combinant les équations (5.37) et (5.39), on déduit que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\, k \in K, \forall\, (u, v) \in A^k \setminus \{a^*\}. \quad (5.40)$$

Considérons maintenant une demande de K^{a^*}, disons $\bar{k}$. Soit $\hat{v}$ un sommet de $V \setminus \{o^{\bar{k}}, d^{\bar{k}}, u^*, v^*\}$. Considérons l'ensemble $B^4 = \{(o^{\bar{k}}, u^*), (u^*, \hat{v}), (\hat{v}, v^*), (v^*, d^{\bar{k}})\}$. Le graphe induit par $T^4 \setminus B^4$ est p-arc connexe. La proposition 5.5 implique qu'il existe une solution, disons (L^{12}, T^4), réalisable pour le 1-PRLPA unitaire telle que $L^{12}_{\bar{k}} = \{(o^{\bar{k}}, u^*), (u^*, \hat{v}), (\hat{v}, v^*), (v^*, d^{\bar{k}})\}$. De plus, comme $a^* \notin T^4$, alors le vecteur d'incidence $(x^{L^{12}}, y^{T^4})$ appartient à F_{a^*}. Considérons maintenant la solution (L^{13}, A) où $L^{13}_k = L^{12}_k$ pour tout $k \in K \setminus \{\bar{k}\}$ et $L^{13}_{\bar{k}} = \{(o^{\bar{k}}, u^*), (u^*, v^*), (v^*, d^{\bar{k}})\}$. La contrainte de capacité associée à l'arc a^* est satisfaite à l'égalité par le vecteur d'incidence $(x^{L^{13}}, y^A)$

puisque $y_{a^*} = 1$ et $x^{\bar{k}}_{a^*} = 1$. On a alors $x^{L^{12}} + y^{T^4} = x^{L^{13}} + y^A$, ce qui implique que $r^{\bar{k}}_{(u^*,\hat{v})} + r^{\bar{k}}_{(\hat{v},v^*)} = s_{(u^*,v^*)} + s_{(v^*,u^*)} + r^{\bar{k}}_{(u^*,v^*)}$. D'après les équations (5.32) et (5.40), il s'ensuit que $r^{\bar{k}}_{(u^*,v^*)} = \lambda^{\bar{k}}_{u^*} - \lambda^{\bar{k}}_{v^*} - \rho$. En réitérant le raisonnement pour toute demande de K^{a^*}, on déduit que

$$r^k_{(u,v)} = \begin{cases} \lambda^k_u - \lambda^k_v - \rho & \text{si } (u,v) = a^*, \\ \lambda^k_u - \lambda^k_v & \text{sinon,} \end{cases} \qquad \forall\, k \in K, \forall\, (u,v) \in A^k. \tag{5.41}$$

Les équations (5.32) et (5.41) impliquent que $(r,s) = \rho(b,c) + (\lambda,\mu)M_{CF}$, ce qui implique que F_{a^*} est une facette. □

Nous nous intéressons maintenant à l'étude polyédrale des contraintes de circuit.

5.1.2.3 Contraintes de circuit

Dans cette partie, nous étudions la structure faciale des contraintes de circuit. Comme le graphe D est complet, pour toute demande $k \in K$, l'arc (o^k, d^k) existe. Les contraintes de circuit sont alors données par les contraintes (4.20). Soit k^* une demande de K et v^* un sommet de $V \setminus \{o^{k^*}, d^{k^*}\}$. Soit $F^{k^*}_{v^*}$ la face induite par l'inégalité de circuit (4.20) associée à la demande k^* et au sommet v^*, c'est-à-dire,

$$F^{k^*}_{v^*} = \{(x,y) \in P_U(D, v_0, K) : x^{k^*}_{(o^{k^*},d^{k^*})} + \sum_{a \in \delta^{\text{out}}(v^*) \cap A^{k^*}} x^{k^*}_a = 1\}.$$

Dans le théorème suivant, on montre que ces contraintes définissent des facettes.

Théorème 5.10 *Les inégalités de circuit* (4.20) *définissent des facettes pour* $P_U(D, v_0, K)$.

Preuve. Soit $b^T x + c^T y \leq \alpha$ l'inégalité de circuit (4.20) correspondant à la demande k^* et au sommet v^*. Soit $r^T x + s^T y \leq \beta$ une inégalité valide définissant une facette F de $P_U(D, v_0, K)$. Supposons que $F^{k^*}_{v^*} \subseteq F$. On montre qu'il existe $\rho \in \mathbb{R}$, $\lambda = (\lambda^k_v, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r,s) = \rho(b,c) + (\lambda,\mu)M_{CF}$.

La proposition 5.2 assure qu'il existe une solution, disons (L^1, T^1), avec $T^1 = \delta(o^{k^*})$. De plus, on a $L^1_{k^*} = \{(o^{k^*}, d^{k^*})\}$. Le vecteur d'incidence (x^{L^1}, y^{T^1}) vérifie alors à l'égalité la contrainte de circuit associée à la demande k^* et au sommet v^*, ce qui implique qu'il appartient à $F^{k^*}_{v^*}$. Soient u et v deux sommets de $V \setminus \{o^{k^*}\}$. La solution (L^1, T^2), avec

$T^2 = T^1 \cup \{(u,v),(v,u)\}$, appartient également à $F^{k^*}_{v^*}$. On a alors $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^2}$, ce qui implique que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V \setminus \{o^{k^*}\}. \tag{5.42}$$

Considérons maintenant trois sommets distincts u, v et w de $V \setminus \{o^{k^*}\}$. La solution (L^1, T^3), avec $T^3 = T^1 \cup \{(u,v),(v,w),(w,u)\}$, appartient aussi à $F^{k^*}_{v^*}$. On a donc $x^{L^1} + y^{T^1} = x^{L^1} + y^{T^3}$, ce qui nous permet de déduire que

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall\ u \neq v \neq w \neq u \in V \setminus \{o^{k^*}\}. \tag{5.43}$$

Il est facile de voir que les solutions (L^1, T^1), (L^1, T^2) et (L^1, T^3) appartiennent toujours à $F^{a^*}_{v^*}$ si l'on considère les sommets d^{k^*} ou v^* à la place du sommet o^{k^*}. En réitérant le même raisonnement en considérant successivement les sommets d^{k^*} et v^*, on obtient, en combinant avec les équations (5.42) et (5.43)

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in V, \tag{5.44}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \begin{array}{l} \forall\ u \neq v \neq w \neq u \in V, \\ \{u,v,w\} \neq \{o^{k^*}, d^{k^*}, v^*\}. \end{array} \tag{5.45}$$

Soit $\bar{v}$ un sommet de $V \setminus \{o^{k^*}, d^{k^*}, v^*\}$. Posons $\mu_{\bar{v}} = 0$ et $\mu_v = -s_{(\bar{v},v)}$ pour tout $v \in V \setminus \{\bar{v}\}$. On a alors $s_{(\bar{v},v)} = \mu_{\bar{v}} - \mu_v$ pour tout $v \in V \setminus \{\bar{v}\}$. Les équations (5.44) impliquent que $s_{(v,\bar{v})} = \mu_v - \mu_{\bar{v}}$ pour tout $v \in V \setminus \{\bar{v}\}$. Soient u et v deux sommets distincts de $V \setminus \{\bar{v}\}$. Par les équations (5.30), on obtient $s_{(\bar{v},u)} + s_{(u,v)} + s_{(v,\bar{v})} = 0$. Comme u et v sont deux sommets quelconques et d'après les valeurs de s associées aux arcs incidents à $\bar{v}$, on obtient

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u,v) \in A. \tag{5.46}$$

Le vecteur s n'est donc rien d'autre que la combinaison linéaire des équations de flot (4.11) associées à y.

On s'intéresse maintenant au vecteur r. Considérons une demande de $K \setminus \{k^*\}$. On suppose, sans perte de généralité, que cette demande correspond à la demande 1. On montre alors que le vecteur r^1 associé à la demande 1 n'est rien d'autre qu'une combinaison linéaire des contraintes de conservation de flot (4.12) associées à la demande 1. Soit (u,v) un arc de $A^1 \setminus \{\delta^{\text{out}}(o^1)\}$. On suppose, sans perte de généralité, qu'il existe une demande, disons la demande 2, ayant pour origine u et destination v. Considérons la solution (L^2, A) telle que

$$L^2_k = \begin{cases} \{(o^1,u),(u,v),(v,d^1)\} & \text{si } k = 1, \\ \{(u,o^1),(o^1,d^1),(d^1,v)\} & \text{si } k = 2, \text{ et } k^* \neq 2, \text{ ou si } k = k^* = 2 \text{ et } v^* \in \{o^1, d^1\}, \\ \{(u,v^*),(v^*,v)\} & \text{si } k = k^* = 2 \text{ et } v^* \notin \{o^1, d^1\}, \\ \{(o^k,d^k)\} & \text{sinon.} \end{cases}$$

L'ensemble $L^2 = \{L^2_k : k \in K\}$ est donc composé de p chemins élémentaires arc-disjoints. De plus, le graphe induit par $A \setminus L$ est clairement fortement connexe, ce qui implique que la solution (L^2, A) est réalisable. Dans tous les cas de figure, le vecteur d'incidence de (L^2, A) satisfait à l'égalité la contrainte de circuit associée à k^* et v^*, ce qui implique que (L^2, A) appartient à $F^{k^*}_{v^*}$. Par ailleurs, comme (o^1, v) n'appartient à aucun chemin de L^2, on peut construire une deuxième solution (L^3, A) telle que

$$L^3_k = \begin{cases} \{(o^1, v), (v, d^1)\} & \text{si } k = 1, \\ L^2_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Comme u et v sont deux sommets quelconques de $V \setminus \{o^1\}$, ceci permet de déduire que

$$r^1_{(o^1,u)} + r^1_{(u,v)} = r^1_{(o^1,v)} \qquad \forall\ (u, v) \in A^1 \setminus \delta^{\text{out}}(o^1). \tag{5.47}$$

Posons $\lambda^1_{o^1} = 0$ et $\lambda^1_v = -r^1_{(o^1,v)}$ pour tout $v \in V \setminus \{o^1\}$. On a alors $r^1_{(o^1,v)} = \lambda^1_{o^1} - \lambda^1_v$ pour tout $v \in V \setminus \{o^1\}$. Considérons un arc (u, v) de $A^1 \setminus \delta^{\text{out}}(o^1)$. D'après (5.47) et les valeurs de r^1 associées aux arcs sortant de o^1, on obtient $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. Comme la demande 1 est une demande quelconque de K différente de k^*, il s'ensuit que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v, \qquad \forall k \in K \setminus \{k^*\}, \forall\ (u, v) \in A^k. \tag{5.48}$$

Considérons maintenant la demande k^*. Soient u et v deux sommets de $V \setminus \{o^{k^*}, d^{k^*}\}$. Supposons d'abord que $u \neq v^*$. Considérons l'ensemble d'arcs $B^1 = \{(o^{k^*}, u), (u, v), (v, v^*), (v^*, d^{k^*}),\ (o^{k^*}, v)\}$. Le graphe induit par $A \setminus B$ est p-arc connexe. La proposition 5.5 implique qu'il existe deux solutions réalisables, disons (L^4, A) et (L^5, A), telles que $L^4_{k^*} = \{(o^{k^*}, u), (u, v),\ (v, v^*), (v^*, d^{k^*})\}$, $L^5_{k^*} = \{(o^{k^*}, v), (v, v^*), (v^*, d^{k^*})\}$ et $L^4_k = L^5_k$ pour tout $k \neq k^*$. Comme v^* appartient aux chemins $L^4_{k^*}$ et $L^5_{k^*}$, la contrainte de circuit associée à k^* et v^* est satisfaite à l'égalité par les vecteurs d'incidence des deux solutions, ce qui implique que ces deux vecteurs appartiennent à $F^{k^*}_{v^*}$. On a alors

$$r^{k^*}_{(o^{k^*},u)} + r^{k^*}_{(u,v)} = r^{k^*}_{(o^{k^*},v)}, \qquad \forall\ (u, v) \in A^{k^*} \setminus \{\delta^{\text{out}}(o^{k^*}) \cup \delta^{\text{in}}(d^{k^*}) \cup \delta^{\text{out}}(v^*)\}. \tag{5.49}$$

Supposons maintenant que $v \neq v^*$. En considérant l'ensemble d'arcs $B^2 = \{(o^{k^*}, v^*), (v^*, u), (u, v), (v, d^{k^*}), (u, d^{k^*})\}$ à la place de B^1 dans la preuve précédente, on montre, de manière similaire, que

$$r^{k^*}_{(u,v)} + r^{k^*}_{(v,d^{k^*})} = r^{k^*}_{(u,d^{k^*})}, \qquad \forall\ (u, v) \in A^{k^*} \setminus \{\delta^{\text{out}}(o^{k^*}) \cup \delta^{\text{in}}(d^{k^*}) \cup \delta^{\text{in}}(v^*)\}. \tag{5.50}$$

Finalement, considérons l'ensemble $B^3 = \{(o^{k^*}, d^{k^*}), (o^{k^*}, v^*), (v^*, d^{k^*})\}$. Le graphe induit par $A \setminus B^3$ est clairement p-arc connexe, ce qui implique, d'après la proposition 5.5, qu'il existe deux solutions appartenant à $F^{k^*}_{v^*}$, disons (L^6, A) et (L^7, A), telles

que $L^6_{k^*} = \{(o^{k^*}, d^{k^*})\}$, $L^7_{k^*} = \{(o^{k^*}, v^*), (v^*, d^{k^*})\}$ et $L^6_k = L^7_k$ pour tout $k \neq k^*$. On obtient alors

$$r^{k^*}_{(o^{k^*}, d^{k^*})} = r^{k^*}_{(o^{k^*}, v^*)} + r^{k^*}_{(v^*, d^{k^*})}. \tag{5.51}$$

Posons $\lambda^{k^*}_{o^{k^*}} = 0$ et $\lambda^{k^*}_v = -r^{k^*}_{(o^{k^*}, v)}$ pour tout $v \in V \setminus \{o^{k^*}, d^{k^*}\}$. On a alors $r^{k^*}_{(o^{k^*}, v)} = \lambda^{k^*}_{o^{k^*}} - \lambda^{k^*}_v$ pour tout $v \in V \setminus \{o^{k^*}, d^{k^*}\}$. Soit (u, v) un arc quelconque de $A^{k^*} \setminus \delta^{\text{out}}(v^*)$, avec u et v deux sommets de $V \setminus \{o^{k^*}, d^{k^*}\}$. Les équations (5.49) et les valeurs de r^{k^*} associées aux arcs sortant de o^{k^*} impliquent que

$$r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v \qquad \forall\ (u, v) \in A^{k^*} \setminus \{\delta^{\text{in}}(d^{k^*}) \cup \delta^{\text{out}}(v^*)\}. \tag{5.52}$$

Soit $\tilde{v}$ un sommet de $V \setminus \{o^{k^*}, d^{k^*}, v^*\}$. Posons $\lambda^{k^*}_{d^*} = -r^{k^*}_{(o^{k^*}, \tilde{v})} - r^{k^*}_{(\tilde{v}, d^{k^*})}$. On a alors $r^{k^*}_{(\tilde{v}, d^{k^*})} = \lambda^{k^*}_{\tilde{v}} - \lambda^{k^*}_{d^*}$. Soit v un sommet de $V \setminus \{o^{k^*}, d^{k^*}, v^*, \tilde{v}\}$. L'équation (5.50) associée à l'arc $(\tilde{v}, v)$ implique que l'on a $r^{k^*}_{(v, d^{k^*})} = \lambda^{k^*}_v - \lambda^{k^*}_{d^{k^*}}$. Comme v est un sommet quelconque différent de o^{k^*}, d^{k^*}, v^* et $\tilde{v}$, on a alors

$$r^{k^*}_{(v, d^{k^*})} = \lambda^{k^*}_v - \lambda^{k^*}_{d^{k^*}} \qquad \forall\ (v, d^{k^*}) \in \delta^{\text{in}}(d^{k^*}) \setminus \{(v^*, d^{k^*}), (o^{k^*}, d^{k^*})\}. \tag{5.53}$$

Posons $\rho = r^{k^*}_{(o^{k^*}, v^*)} + r^{k^*}_{(v^*, d^{k^*})} - r^{k^*}_{(o^{k^*}, \tilde{v})} - r^{k^*}_{(\tilde{v}, d^{k^*})}$. On obtient alors $r^{k^*}_{(v^*, d^{k^*})} = \lambda^{k^*}_{v^*} - \lambda^{k^*}_{d^{k^*}} + \rho$. Soit v un sommet de $V \setminus \{o^{k^*}, d^{k^*}, v^*\}$. L'équation (5.50) associée à l'arc (v^*, v) implique que l'on a $r^{k^*}_{(v^*, v)} = \lambda^{k^*}_v - \lambda^{k^*}_{d^{k^*}} + \rho$. Comme v est un sommet quelconque différent de o^{k^*}, d^{k^*} et v^*, on a alors

$$r^{k^*}_{(v^*, v)} = \lambda^{k^*}_{v^*} - \lambda^{k^*}_v + \rho \qquad \forall\ (v^*, v) \in \delta^{\text{out}}(v^*). \tag{5.54}$$

Finalement, l'équation (5.51) permet de déduire que $r^{k^*}_{(o^{k^*}, d^{k^*})} = \lambda^{k^*}_{o^{k^*}} - \lambda^{k^*}_{d^{k^*}} + \rho$. D'après les équations (5.52)-(5.54), il s'ensuit que

$$r^{k^*}_{(u,v)} = \begin{cases} \lambda^{k^*}_u - \lambda^{k^*}_v + \rho & \text{si } (u, v) \in \delta^{\text{out}}(v^*) \cup \{(o^{k^*}, d^{k^*})\}, \\ \lambda^{k^*}_u - \lambda^{k^*}_v & \text{sinon.} \end{cases} \qquad \forall\ (u, v) \in A^{k^*}. \tag{5.55}$$

Les équations (5.46), (5.48) et (5.55) permettent alors de conclure que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$, ce qui implique que $F^{k^*}_{v^*}$ est une facette. □

5.1.2.4 Contraintes de vulnérabilité renforcées

Dans cette partie, nous étudions la structure faciale des contraintes de vulnérabilité renforcées (4.24). Comme il a été remarqué, les contraintes de vulnérabilité (4.21) sont

dominées par les contraintes (4.24) et ne peuvent donc définir des facettes. Soit W^* un sous-ensemble de sommets de V tel que le dépôt v_0 appartient à W^*, il existe au moins une demande ayant ses deux extrémités dans $\overline{W}^*$ et il n'existe aucune demande ayant exactement une extrémité dans W^*. Soit F_{W^*} la face induite par la contrainte de vulnérabilité renforcée (4.24) associée à W^*, c'est-à-dire,

$$F_{W^*} = \{(x,y) \in P_U(D, v_0, K) : \sum_{a \in \delta^{\text{out}}(W^*)} y_a - \sum_{k \in A_\Phi(\overline{W}^*)} \sum_{a \in \delta^{\text{out}}(W^*) \cap A^k} x_a^k = 1\}.$$

On donne dans le théorème suivant les conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

Théorème 5.11 *Les inégalités* (4.24) *définissent des facettes pour* $P_U(D, v_0, K)$ *si et seulement si* $|\overline{W}^*| \geq 3$.

Preuve. Tout d'abord, supposons que $|\overline{W}^*| = 2$. Dans ce cas, comme $A_\Phi(\overline{W}^*) \neq \emptyset$, on suppose, sans perte de généralité, que $\overline{W}^* = \{o^1, d^1\}$. Si une demande de $K \setminus \{1\}$, disons $\bar{k}$, est transportée sur l'arc (d^1, o^1), il existe alors un arc sortant de o^1 sur lequel est transportée $\bar{k}$. De plus, il existe un deuxième arc sortant de o^1 sur lequel la demande 1 est transportée. Comme le graphe induit par y est Eulérien, on a alors $y(\delta^{\text{in}}(o^1)) \geq 2$, ce qui implique qu'il existe un arc a de $\delta^{\text{in}}(o^1) \cap \delta^{\text{out}}(W^*)$ avec $y_a = 1$. Or, la demande 1 n'est pas transportée ni sur cet arc, ni sur l'arc utilisé pour transporter $\bar{k}$ de W^* à d^1. On a alors $y(\delta^{\text{out}}(W^*)) - x^1(\delta^{\text{out}}(W^*) \cap A^1) \geq 2$. L'inégalité de vulnérabilité renforcée est alors dominée par l'inégalité valide

$$y(\delta^{\text{out}}(W^*)) - x^1(\delta^{\text{out}}(W^*) \cap A^1) - \sum_{k \in K \setminus \{1\}} x^k_{(d^1,o^1)} \geq 1.$$

Comme cette équation définit une face propre, la contrainte de vulnérabilité renforcée (4.24) associée à W^* ne définit donc pas une facette.

Dans le reste de la preuve, on suppose que $|\overline{W}^*| \geq 3$. De plus, on suppose que la demande 1 a ses deux extrémités dans $\overline{W}^*$. Soit $b^T x + c^T y \geq \alpha$ l'inégalité (4.24) associée au sous-ensemble de sommets W^*. Soit $r^T x + s^T y \geq \beta$ une inégalité valide définissant une facette F de $P_U(D, v_0, K)$. Supposons que $F_{W^*} \subseteq F$. On montre qu'il existe $\rho \in \mathbb{R}$, $\lambda = (\lambda_v^k, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = \rho(b, c) + (\lambda, \mu) M_{CF}$.

Soient $\bar{v}$ un sommet de W^* et $\tilde{v}$ un sommet de $\overline{W}^*$. Considérons la solution (L^1, T^1) définie par

$$L_k^1 = \begin{cases} \{(o^k, \bar{v}), (\bar{v}, d^k)\} & \text{si } o^k, d^k \in W^*, \\ \{(o^k, \tilde{v}), (\tilde{v}, d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$T^1 = \{(\bar{v}, v), (v, \bar{v}) : v \in W^* \setminus \{\bar{v}\}\} \cup \{(\tilde{v}, v), (v, \tilde{v}) : v \in \overline{W}^* \setminus \{\tilde{v}\}\} \cup \{(\bar{v}, \tilde{v}), (\tilde{v}, \bar{v})\}.$$

On montre que la solution (L^1, T^1) est réalisable. Tout d'abord, il est facile de voir que L^1 est un ensemble de $o^k d^k$-chemins élémentaires arc-disjoints, $k \in K$. Posons

$$\Gamma_1 = \{(\bar{v}, o^k) : k \in A_\Phi(W^*)\} \cup \{(\tilde{v}, o^k) : k \in A_\Phi(\overline{W}^*)\} \cup \{(v_0, \bar{v}), (\bar{v}, \tilde{v})\}.$$

Γ_1 correspond à une v_0-arborescence du graphe induit par T^1 couvrant toutes les origines des demandes de K et n'intersectant aucun arc de L^1. La réalisabilité de la solution (L^1, T^1) est alors donnée par la proposition 5.1. Comme T^1 ne contient qu'un seul arc sortant de W^*, le vecteur d'incidence (x^{L^1}, y^{T^1}) appartient à F_{W^*}.
Supposons que $|W^*| \geq 3$. Il existe alors deux sommets de $W^* \setminus \{\bar{v}\}$, disons u et v. Considérons la solution $(L^1, \bar{T}^1)$ avec $\bar{T}^1 = T^1 \cup \{(u, v), (v, u)\}$. $(x^{L^1}, y^{\bar{T}^1})$ appartient aussi à F_{W^*}, ce qui implique que $x^{L^1} + y^{T^1} = x^{L^1}, y^{\bar{T}^1}$. On déduit alors que $s_{(u,v)} + s_{(v,u)}$. Comme les sommets u, v et $\bar{v}$ sont des sommets quelconques de W^*, on obtient

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in W^*. \tag{5.56}$$

Le résultat reste vrai si $|W^*| = 2$, *i.e.*, $W^* = \{v_0, v_1\}$. En effet, considérons la solution $(L^1, \tilde{T}^1)$ avec $\tilde{T}^1 = T^1 \setminus \{(v_0, v_1), (v_1, v_0)\}$. Le vecteur d'incidence de cette solution appartient également à F_{W^*} car il n'existe aucune demande ayant ses deux extrémités dans W^*. Comme les vecteurs d'incidence des solutions (L^1, T^1) et $(L^1, \tilde{T}^1)$ vérifient l'inégalité $b^T x + c^T y \geq \alpha$ à l'égalité, on déduit que $s_{(v_0,v_1)} + s_{(v_1,v_0)} = 0$.

Supposons maintenant que $|W^*| \geq 4$. Dans ce cas, il existe trois sommets de $V \setminus \{\bar{v}\}$, disons u, v et w. Le vecteur d'incidence de la solution $(L^1, \hat{T}^1)$, avec $\hat{T}^1 = T^1 \cup \{(u, v), (v, w), (w, u)\}$, appartient également à F_{W^*}. Comme les sommets u, v, w et $\bar{v}$ sont choisis arbitrairement, il s'ensuit que

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall\ u \neq v \neq w \neq u \in W^*. \tag{5.57}$$

Supposons maintenant que $|W^*| = 3$. Sans perte de généralité, on suppose que $W^* = \{v_0, o^2, d^2\}$. Considérons la solution (L^2, T^2) définie par

$$L^2_k = \begin{cases} \{(o^2, d^2)\} & \text{si } k = 2, \\ L^1_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$T^2 = \{(v_0, o^2), (o^2, d^2), (d^2, v_0), (v_0, \tilde{v}), (\tilde{v}, v_0)\} \cup \{(\tilde{v}, v), (v, \tilde{v}) : v \in \overline{W}^*\}.$$

Il est facile de voir que le vecteur d'incidence de (L^2, T^2) appartient à F_{W^*}. Soit $\bar{T}^2 = T^2 \cup \{(v_0, d^2), (d^2, o^2), (o^2, v_0)\}$. Le vecteur d'incidence $(x^{L^2}, y^{\bar{T}^2})$ appartient aussi à F_{W^*}. On a alors

$$s_{(v_0,d^2)} + s_{(d^2,o^2)} + s_{(o^2,v_0)} = 0. \tag{5.58}$$

Le sommet $\tilde{v}$ est un sommet quelconque de $\overline{W}^*$. On peut donc appliquer le même raisonnement que précédemment en considérant cette fois que les sommets u, v et w considérés précédemment sont des sommets de $\overline{W^*}$. On déduit alors que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall\ u \neq v \in \overline{W}^*, \tag{5.59}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall\ u \neq v \neq w \neq u \in W^*. \tag{5.60}$$

On détermine maintenant certaines propriétés du vecteur s associé aux arcs de la coupe $\delta(W^*)$. On suppose, sans perte de généralité, que $|W^*| \geq 3$. Les résulats pour le cas $|W^*| \leq 2$ peuvant être obtenus de manière similaire, la preuve ne sera pas donnée. Soit u et v deux sommets distincts quelconques de $W^* \setminus \{\bar{v}\}$. (On rappelle que $\bar{v}$ et $\tilde{v}$ appartiennent respectivement à W^* et $\overline{W}^*$.) Considérons la solution (L^1, T^3) avec $T^3 = A(W^*) \cup A(\overline{W}^*) \cup \{(u,\tilde{v}),(\tilde{v},v)\} \setminus \{(u,v)\}$. Notons par D^3 le graphe induit par T^3. On remarque que D^3 est Eulérien. Les sous-graphes de D^3 respectivement induits par les sous-ensembles de sommets $W^* \cup \{\tilde{v}\} \setminus \{\bar{v}\}$ et $\overline{W}^* \setminus \{\tilde{v}\}$ sont fortement connexes. De plus, comme $|W^*| \geq 3$ et $|\overline{W}| \geq 3$, les sommets $\bar{v}$ et $\tilde{v}$ possèdent chacun au minimum un arc sortant et un arc entrant dans D^3 n'appartenant pas à L^1. On déduit alors que le graphe $D^3 \setminus L^1$ est fortement connexe. Il existe donc dans $D^3 \setminus L^1$ une v_0-arborescence couvrante n'intersectant aucun arc de L^1. D'après la proposition 5.1, la solution (L^1, T^3) est donc réalisable. Comme T^3 ne contient qu'un seul arc de la coupe sortante de W^*, le vecteur d'incidence de cette solution appartient à F_{W^*}. Considérons maintenant la solution $(L^1, \bar{T}^3)$, avec $\bar{T}^3 = T^3 \cup \{(u,v),(v,\tilde{v})\} \setminus \{(u,\tilde{v})\}$. Le vecteur $(x^{L^1}, y^{\bar{T}^3})$ est un point de F_{W^*}. (La preuve est similaire à celle utilisée pour montrer que (x^{L^1}, y^{T^3}) appartient à F_{W^*}.) Comme les vecteurs d'incidence de (L^1, T^3) et $(L^1, \bar{T}^3)$ vérifient à l'égalité $r^T x + s^T y \geq \beta$, il s'ensuit que $s_{(u,\tilde{v})} = s_{(u,v)} + s_{(v,\tilde{v})}$. Comme $\bar{v}$, $\tilde{v}$ sont choisis arbitrairement et u et v sont deux sommets distincts quelconques de $W^* \setminus \{\bar{v}\}$, on obtient

$$s_{(u,w)} = s_{(u,v)} + s_{(v,w)} \qquad \forall\ u \neq v \in W^*, \forall\ w \in \overline{W}^*. \tag{5.61}$$

En utilisant le raisonnement précédent en considérant cette fois que u et v sont deux sommets quelconques de $\overline{W}^*$, on déduit alors le résultat suivant

$$s_{(u,w)} = s_{(u,v)} + s_{(v,w)} \qquad \forall\ u \neq v \in \overline{W^*}, \forall\ w \in W^*. \tag{5.62}$$

D'une manière similaire, on peut montrer que les équations (5.61) (respectivement (5.62)) restent vraies si v est un sommet de $\overline{W}^*$ (respectivement W^*) différent de w. On déduit alors, avec les équations (5.61) et (5.62), que

$$s_{(u,w)} = s_{(u,v)} + s_{(v,w)} \qquad \forall\ (u,w) \in \delta(W^*), \forall\ v \in V \setminus \{u,w\}. \tag{5.63}$$

Posons $\mu_{v_0} = 0$ et $\mu_v = -s_{(v_0,v)}$ pour tout $v \in W^* \setminus \{v_0\}$. On a alors $s_{(v_0,v)} = \mu_{v_0} - \mu_v$ pour tout $v \in W^* \setminus \{v_0\}$. Par les contraintes (5.56), on déduit que $s_{(v,v_0)} = \mu_v - \mu_{v_0}$ pour tout $v \in W^* \setminus \{v_0\}$. Considérons un arc (u, v) de $A(W^*)$ tel que u et v soient différents de v_0. D'après les contraintes (5.57), ou (5.58) si $|W^*| = 3$, et d'après les valeurs de s associées aux arcs sortant de v_0, on obtient que $s_{(u,v)} = \mu_u - \mu_v$. Il en résulte que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u, v) \in A(W^*). \tag{5.64}$$

Posons maintenant $\mu_{\tilde{v}} = s_{(\tilde{v},v_0)}$ et $\mu_v = -s_{(\tilde{v},v)} + s_{(\tilde{v},v_0)}$ pour tout $v \in \overline{W}^* \setminus \{\tilde{v}\}$. On déduit que $s_{(\tilde{v},v_0)} = \mu_{\tilde{v}} - \mu_{v_0}$ et $s_{(\tilde{v},v)} = \mu_{\tilde{v}} - \mu_v$ pour tout $v \in \overline{W}^* \setminus \{\tilde{v}\}$. Par les contraintes (5.59), $s_{(v,\tilde{v})} = \mu_v - \mu_{\tilde{v}}$ pour tout $v \in \overline{W}^* \setminus \{\tilde{v}\}$. Étant donné un arc (u, v) de $A(\overline{W}^*)$ n'appartenant pas à $\delta(\tilde{v})$, on obtient, par les contraintes (5.60), $s_{(u,v)} = \mu_u - \mu_v$. Il s'ensuit alors que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u, v) \in A(\overline{W}^*). \tag{5.65}$$

Considérons maintenant un sommet v de $\overline{W^*}$ différent de $\tilde{v}$. D'après l'équation (5.63) associée à l'arc (v, v_0) et au sommet $\tilde{v}$ et les valeurs de s associées à $(v, \tilde{v})$ et $(\tilde{v}, v_0)$, on déduit que $s_{(v,v_0)} = \mu_v - \mu_{v_0}$ pour tout $v \in \overline{W}^*$. Considérons maintenant un arc (u, v) de $\delta^{\text{in}}(W^*)$ avec $v \neq v_0$. Par l'équation (5.63) associée à l'arc (u, v) et au sommet v_0, on obtient $s_{(u,v)} = \mu_u - \mu_v$. Comme (u, v) est un arc quelconque de la coupe entrante de W^*, il s'ensuit que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u, v) \in \delta^{\text{in}}(W^*). \tag{5.66}$$

Posons $\rho = s_{(v_0,\tilde{v})} + s_{(\tilde{v},v_0)}$. On a alors $s_{(v_0,\tilde{v})} = \mu_{v_0} - \mu_{\tilde{v}} + \rho$. Considérons un sommet v de $\overline{W}^* \setminus \{\tilde{v}\}$. L'équation (5.63) associée à l'arc (v_0, v) et au sommet $\tilde{v}$ implique d'après les valeurs $s_{(v_0,\tilde{v})}$ et $s_{(\tilde{v},v)}$, que $s_{(v_0,v)} = \mu_{v_0} - \mu_v + \rho$. Considérons maintenant un arc (u, v) de $\delta^{\text{out}}(W^*)$ tel que $u \neq v_0$. D'après (5.63) associée à (u, v) et v_0, on déduit que $s_{(u,v)} = \mu_u - \mu_v + \rho$. On peut alors généraliser ce résultat pour tout arc sortant de W^*. En considérant également les équations (5.64)-(5.66), on obtient

$$s_{(u,v)} = \begin{cases} \mu_u - \mu_v + \rho & \text{si } (u, v) \in \delta^{\text{out}}(W^*), \\ \mu_u - \mu_v & \text{sinon,} \end{cases} \qquad \forall\ (u, v) \in A. \tag{5.67}$$

On détermine maintenant les valeurs du vecteur r associées aux demandes ayant leurs deux extrémités dans W^*. Considérons une demande de $A_\Phi(W^*)$, disons la demande 2. Soit (u, v) un arc de $(A(W^*) \cap A^2) \setminus \delta^{\text{out}}(o^2)$. Considérons la solution (L^4, T^4) définie par

$$L^4_k = \begin{cases} \{(o^2, u), (u, v), (v, d^2)\} & \text{si } k = 2, \\ \{(u, v_0), (v_0, v)\} & \text{si } k = (u, v), \\ \{(o^k, d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$T^4 = A(W^*) \cup A(\overline{W}^*) \cup \{(v_0, o^1), (o^1, v_0)\}.$$

Comme T^4 induit un graphe Eulérien et L^4_k, $k \in K$ correspond à un $o^k d^k$-chemin, le vecteur d'incidence (x^{L^4}, y^{T^4}) satisfait les contraintes (4.10)-(4.14). Comme les chemins des demandes sont arc-disjoints et utilisent uniquement des arcs de T^4, les contraintes de capacité (4.15) sont aussi satisfaites par (x^{L^4}, y^{T^4}). Par ailleurs, considérons n'importe quel sous-ensemble W de V tel que $v_0 \in W$, $A_\Phi(\overline{W}) \neq \emptyset$ et $\delta(W) = \emptyset$. La définition des chemins L^4_k, $k \in K$ implique que

$$\sum_{k \in A_\Phi(\overline{W})} x^k(\delta^{\text{out}}(W) \cap A^k) \begin{cases} = 0 & \text{si } 2, (u,v) \in A_\Phi(W), \\ \leq 1 & \text{sinon.} \end{cases}$$

Supposons que $W^* \subseteq W$. Dans ce cas, les demandes 2 et (u, v) appartiennent à $A_\Phi(W)$. Comme le graphe est connexe, la contrainte de vulnérabilité renforcée (4.24) associée à W est satisfaite par (x^{L^4}, y^{T^4}). Si $W^* \nsubseteq W$, alors l'ensemble $U = W^* \setminus W$ n'est pas vide. Comme il existe au minimum trois sommets dans W^* et T^4 contient le graphe complet induit par W^*, alors $y(\delta^{\text{in}}(U)) \geq 2$, ce qui implique que $y(\delta^{\text{out}}(W)) \geq 2$. Par conséquent, la contrainte de vulnérabilité renforcée est satisfaite par (x^{L^4}, y^{T^4}). De plus, T^4 intersecte seulement un arc de $\delta^{\text{out}}(W^*)$ ce qui implique dans les deux cas que (x^{L^4}, y^{T^4}) appartient à F_{W^*}. Par ailleurs, l'arc (o^2, v) appartient à T^4 mais n'appartient à aucun chemin de L^4. On peut alors construire une autre solution $(\bar{L}^4, T^4)$ de F_{W^*} où $\hat{L}^4$ est défini par

$$\bar{L}^4_k = \begin{cases} \{(o^2, v), (v, d^2)\} & \text{si } k = 2, \\ L^4_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

On déduit alors que $r^2_{(o^2,u)} + r^2_{(u,v)} = r^2_{(o^2,v)}$. Comme (u, v) est un arc choisi arbitrairement, on peut étendre ce résultat comme suit

$$r^2_{(o^2,u)} + r^2_{(u,v)} = r^2_{(o^2,v)} \qquad \forall\ (u,v) \in A^2 \cap A(W^*), u \neq o^2. \tag{5.68}$$

Considérons un arc (u, v) de $A(W^*) \cap A^2$. Dans la suite, on suppose que (u, v) est différent de (o^2, d^2). La preuve lorsque (u, v) est égal à (o^2, d^2) est similaire et sera donc omise. Considérons l'ensemble d'arcs $\tilde{A}$ donné par

$$\tilde{A} = \begin{cases} \{(u, o^2), (o^2, v)\} & \text{si } v = d^2, \\ \{(u, d^2), (d^2, v)\} & \text{sinon.} \end{cases}$$

Soit w un sommet quelconque de $\overline{W}^*$. Considérons l'ensemble d'arcs $T^5 = (A(W^*) \setminus \tilde{A}) \cup A(\overline{W}^*) \cup \{(u, w), (w, v)\}$. Le graphe orienté induit par T^5 est Eulérien. Soit L^5 tel

que

$$L_k^5 = \begin{cases} \{(o^2,u),(u,v),(v,d^2)\} & \text{si } k=2, \\ \{(u,v_0),(v_0,v)\} & \text{si } k=(u,v), \\ \{(o^k,d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

La solution (L^5, T^5) est réalisable pour le 1-PRLPA unitaire. En effet, considérons la v_0-arborescence Γ_2 définie par $\Gamma_2 = \{(v_0,o^k) : k \in A_\Phi(W^*)\} \cup \{(v_0,u),(u,w)\} \cup \{(w,o^k) : k \in A_\Phi(\overline{W}^*)\}$. Si $u = v_0$ et $v = d^2$, remplaçons dans Γ_2 l'arc (v_0,o^2) par les deux arcs (w,d^2) et (d^2,o^2). Par conséquent, Γ_2 utilise uniquement des arcs de T^5 et couvre toutes les origines de K. De plus, aucun chemin de L^5 intersecte Γ_2, ce qui implique, par la proposition 5.1, que la solution est réalisable.
Considérons maintenant la solution $(\bar{L}^5, T^5)$ avec

$$\bar{L}_k^5 = \begin{cases} \{(o^2,u),(u,w),(w,v),(v,d^2)\} & \text{si } k=2, \\ L_k^5 & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

En utilisant la v_0-arborescence Γ_2 (dans laquelle l'arc (v_0,d^2) est ajouté si $(u,v) = (v_0,d^2)$), on obtient, d'après la proposition 5.1, que $(\bar{L}^5, T^5)$ est réalisable. Comme la contrainte de vulnérabilité renforcée associée à W^* est satisfaite à l'égalité pour les deux solutions (L^5,T^5) et $(\bar{L}^5,T^5)$, ces dernières appartiennent à F_{W^*}. On déduit alors que $r^2_{(u,v)} = r^2_{(u,w)} + r^2_{(w,v)}$. On peut généraliser ce résultat en appliquant ce raisonnement pour tout arc (u,v) de $A(W^*) \cap A^2$ et pour tout sommet w de $\overline{W}^*$, ce qui implique que

$$r^2_{(u,v)} = r^2_{(u,w)} + r^2_{(w,v)} \qquad \forall\ (u,v) \in A(W^*) \cap A^2, \forall\ w \in \overline{W}^*. \tag{5.69}$$

On s'intéresse maintenant aux arcs de $A(\overline{W}^*)$. Pour cela, considérons deux sommets distincts de $\overline{W}^*$, disons $\tilde{v}$ et $\hat{v}$. Considérons l'ensemble $T^6 = A(\overline{W}^*) \setminus \{(\hat{v},\tilde{v})\} \cup \{(v_0,v),(v,v_0) : v \in W^*\} \cup \{(\hat{v},d^2),(d^2,o^2),(o^2,\tilde{v})\}$. Il est clair que T^6 induit un graphe Eulérien. Soit w un sommet de $\overline{W}^*$ différent de $\tilde{v}$ et $\hat{v}$. Soit L^6 un ensemble de chemins élémentaires défini par

$$L_k^6 = \begin{cases} \{(o^k,w),(w,d^k)\} & \text{si } k \in A_\Phi(\overline{W}^*), \\ \{(o^k,v_0),(v_0,d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Les chemins de L^6 utilisent uniquement des arcs de T^6. Considérons l'ensemble d'arcs $\Gamma_3 = \{(\tilde{v},v) : v \in \overline{W}^* \setminus \{w\}\}$. Par la définition de L^6, il existe un arc entrant dans w n'appartenant pas à un chemin de L^6. Ajoutons cet arc dans Γ_3. L'ensemble d'arcs Γ_3 correspond alors à une $\tilde{v}$-arborescence couvrant tous les sommets de $\overline{W}^*$ et n'intersectant pas L^6. Soit $\Gamma_4 = \Gamma_3 \cup \{(v_0,o^k) : k \in A_\Phi(W^*)\} \cup \{(o^2,\tilde{v})\}$. Γ_4 correspond

alors à une v_0-arborescence couvrant toutes les origines des demandes de K et n'intersectant pas L^6. Comme L^6 correspond à un ensemble de chemins arc-disjoints, d'après la proposition 5.1, (L^6, T^6) est réalisable. De plus, comme T^6 contient exactement un arc sortant de W^*, le vecteur d'incidence de (L^6, T^6) appartient à F_{W^*}. Considérons $\bar{L}^6 = \{\bar{L}^6_1, \bar{L}^6_2, \ldots, \bar{L}^6_p\}$ tel que

$$\bar{L}^6_k = \begin{cases} \{(o^2, \tilde{v}), (\tilde{v}, \hat{v}), (\hat{v}, d^2)\} & \text{si } k = 2, \\ L^6_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

$\bar{L}^6$ correspond à un ensemble de $o^k d^k$-chemins élémentaires arc-disjoints, $k \in K$. Considérons la solution $(\bar{L}^6, T^6)$. Γ_4 intersecte uniquement le chemin $\hat{L}^6_2$ et l'arc (v_0, o^2) appartient à $\Gamma_4 \setminus \hat{L}^6$. D'après la proposition 5.1, $(\hat{L}^6, T^6)$ est réalisable. De plus, il est clair que cette solution appartient aussi à F_{W^*}. On déduit alors que $r^2_{(o^2,d^2)} = r^2_{(o^2,\tilde{v})} + r^2_{(\tilde{v},\hat{v})} + r_{(\hat{v},d^2)}$. Comme $\tilde{v}$ et $\hat{v}$ sont deux sommets quelconques de $\overline{W}^*$, il s'ensuit que

$$r^2_{(o^2,d^2)} = r^2_{(o^2,u)} + r^2_{(u,v)} + r_{(v,d^2)} \qquad \forall\ u \neq v \in \overline{W}^*. \tag{5.70}$$

Posons $\lambda^2_{o^2} = 0$ et $\lambda^2_v = -r^2_{(o^2,v)}$ pour tout $v \in V \setminus \{o^2\}$. Cela implique que $r^2_{(o^2,v)} = \lambda^2_{o^2} - \lambda_v$ pour tout $v \in V \setminus \{o^2\}$. Considérons un arc (u, v) de $A^2 \cap A(W^*)$ tel que u est différent de o^2. L'équation (5.68) implique que $r^2_{(u,v)} = \lambda^2_u - \lambda^2_v$. D'après la définition de (u, v) et les valeurs de r^2 associées aux arcs sortant de o^2, il s'ensuit que

$$r^2_{(u,v)} = \lambda^2_u - \lambda^2_v \qquad \forall\ (u, v) \in A(W^*) \cap A^2. \tag{5.71}$$

Soit (u, v) un arc entrant de W^*, avec $v \neq o^2$. L'équation (5.69) associé à l'arc (o^2, v) et au sommet u permet de déduire que $r^2_{(u,v)} = \lambda^2_u - \lambda^2_v$ pour tout arc $(u, v) \in \delta^{\text{in}}(W^* \setminus \{o^2\})$. Supposons maintenant que (u, v) est un arc sortant de W^* tel que $u \neq d^2$. Par (5.69) associé à l'arc (u, d^2) et au sommet v, il s'ensuit que $r^2_{(u,v)} = \lambda^2_u - \lambda^2_v$. On obtient alors

$$r^2_{(u,v)} = \lambda^2_u - \lambda^2_v \qquad \forall\ (u, v) \in \delta(W^*) \cap A^2. \tag{5.72}$$

Finalement, considérons n'importe quel arc (u, v) de $A(\overline{W}^*)$. D'après l'équation (5.70), et les valeurs de r^2 associées aux arcs (o^2, u), (v, d^2) et (o^2, d^2), on déduit que $r^2_{(u,v)} = \lambda^2_u - \lambda^2_v$. On obtient alors

$$r^2_{(u,v)} = \lambda^2_u - \lambda^2_v \qquad \forall\ (u, v) \in A^2. \tag{5.73}$$

Comme la demande 2 est une demande quelconque de $A_\Phi(W^*)$, en réitérant le raisonnement pour chaque demande de cet ensemble, il en résulte que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\ k \in A_\Phi(W^*), \forall\ (u, v) \in A^k. \tag{5.74}$$

On considère maintenant le vecteur r associé aux demandes de $A_\Phi(\overline{W}^*)$. Pour cela, considérons une demande de $A_\Phi(\overline{W}^*)$, disons la demande 1. Soit (u,v) un arc de $A^1 \cap A(\overline{W}^*) \setminus \delta(o^1)$. Considérons la solution (L^7, T^7) avec

$$L_k^7 = \begin{cases} \{(o^1,u),(u,v),(v,d^1)\} & \text{si } k=1, \\ \{(u,d^1),(d^1,v)\} & \text{si } k=(u,v), \\ \{(o^k,d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$T^7 = A(W^*) \cup A(\overline{W}^*) \cup \{(v,v_0),(v_0,u)\} \setminus \{(v,u)\}.$$

Considérons la v_0-arborescence $\Gamma_5 = \{(v_0,o^k) : k \in A_\Phi(W^*)\} \cup \{(u,o^k) : k \in A_\Phi(\overline{W}^*)\} \cup \{(v_0,u)\}$. Si v est l'origine d'une demande de $A_\Phi(\overline{W}^*)$ alors on remplace dans Γ_5 l'arc (u,v) par les deux arcs (u,d^1) et (d^1,v). La proposition 5.1 implique que la solution est réalisable. De plus, comme la contrainte de vulnérabilité renforcée associée à W^* est vérifiée à l'égalité par (x^{L^7}, y^{T^7}), alors (x^{L^7}, y^{T^7}) appartient à F_{W^*}. Considérons l'ensemble de chemins $\bar{L}^7$ tel que

$$\bar{L}_k^7 = \begin{cases} \{(o^1,v),(v,d^1)\} & \text{si } k=1, \\ L_k^7 & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

La solution $(\bar{L}^7, T^7)$ est aussi une solution réalisable. De plus, la contrainte de vulnérabilité renforcée est vérifiée à l'égalité par son vecteur d'incidence. Il s'ensuit que $(x^{\bar{L}^7}, y^{T^7})$ appartient également à F_{W^*}. On déduit alors que $r^1_{(o^1,u)} + r^1_{(u,v)} = r^1_{(o^1,v)}$. Comme u et v sont choisis arbitrairement, on déduit que

$$r^2_{(o^2,u)} + r^2_{(u,v)} = r^2_{(o^2,v)} \qquad \forall\ (u,v) \in A^1 \cap A(\overline{W}^*) \setminus \delta(o^1). \tag{5.75}$$

Soient (u,v) un arc de $A^1 \cap A(\overline{W}^*)$ et w un sommet de W^*. On peut alors construire une solution $(\tilde{L}^7, \tilde{T}^7)$ avec

$$\tilde{L}_k^7 = \begin{cases} \{(o^1,u),(u,w),(w,v),(v,d^1)\} & \text{si } k=1, \\ L_k^7 & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$\tilde{T}^7 = T^7 \setminus \{(u,v)\} \cup \{(u,w),(w,v)\}.$$

Le point $(x^{\tilde{L}^7}, y^{\tilde{T}^7})$ appartient également à $P_U(D, v_0, K)$. De plus, on a $y(\delta^{\text{out}}(W^*)) = 2$ et $x^1(\delta^{\text{out}}(W^*) \cap A^1) = 1$, ce qui implique que la contrainte de vulnérabilité renforcée associée à W^* est vérifiée à l'égalité. Il s'ensuit que $(x^{\tilde{L}^7}, y^{\tilde{T}^7})$ est un point de F_{W^*}. Comme (u,v) et w sont choisis arbitrairement et comme les points (x^{L^7}, y^{T^7}) et $(x^{\tilde{L}^7}, y^{\tilde{T}^7})$ vérifie à l'égalité la contrainte $r^T x + s^T y \geq \beta$, on obtient alors

$$r^1_{(u,v)} + s_{(u,v)} = r^1_{(u,w)} + r^1_{(w,v)} + s_{(u,w)} + s_{(w,v)} \qquad \begin{array}{l} \forall\ (u,v) \in A^1 \cap A(\overline{W}^*), \\ u \neq o^1, \forall\ w \in W^* \end{array} \tag{5.76}$$

Considérons maintenant un arc (u, v) de $A(W^*)$. On suppose, sans perte de généralité, que $|W^*| \geq 3$. Soit w un sommet de $W^* \setminus \{u, v\}$. Considérons la solution (L^8, T^8) avec

$$L_k^8 = \begin{cases} \{(o^k, w), (w, d^k)\} & \text{si } k \in A_\Phi(W^*), \\ \{(o^k, d^k)\} & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$T^8 = A(W^*) \cup A(\overline{W}^*) \cup \{(u, o^1), (d^1, v)\} \setminus \{(u, v), (d^1, o^1)\}.$$

La solution (L^8, T^8) est réalisable. En effet, considérons l'ensemble $\Gamma_6 = \{(v_0, o^k) : k \in A_\Phi(W^*)\} \cup \{(o^1, o^k) : k \in A_\Phi(\overline{W}^*) \setminus \{1\}\} \cup \{(v_0, u), (u, o^1)\}$. Si $w = v_0$ et s'il existe une demande, disons la demande 2, telle que $u = d^2$, alors on remplace dans Γ_6 l'arc (v_0, o^2) par l'arc (o^2, d^2). De la même manière, si $u = v_0$ et s'il existe une demande, disons la demande 2, telle que $v = o^2$, alors on remplace l'arc (v_0, o^2) par les arcs (v_0, w) et (w, o^2). Γ_6 est une v_0-arborescence couvrant les origines des demandes de K et n'intersectant pas L^8. La réalisabilité de la solution est alors donnée par la proposition 5.1. De plus, (x^{L^8}, y^{T^8}) appartient à F_{W^*}. Considérons maintenant la solution $(\bar{L}^8, \bar{T}^8)$, avec

$$\bar{L}_k^8 = \begin{cases} \{(o^1, u), (u, v), (v, d^1)\} & \text{si } k = 1, \\ L_k^8 & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

et

$$\bar{T}^8 = T^8 \cup \{(o^1, u), (u, v), (v, d^1)\} \setminus \{(o^1, d^1)\}.$$

Le point $(x^{\bar{L}^8}, y^{\bar{T}^8})$ appartient à $P_U(D, v_0, K)$ et vérifie à l'égalité la contrainte de vulnérabilité renforcée associée à W^*, ce qui implique que $(x^{\bar{L}^8}, y^{\bar{T}^8}) \in F_{W^*}$. Comme (u, v) et w sont choisis arbitrairement, on déduit alors que

$$\begin{aligned} r^1_{(o^1,d^1)} + s_{(o^1,d^1)} = \ & r^1_{(o^1,u)} + r^1_{(u,v)} + r^1_{(v,d^1)} \\ & + s_{(o^1,u)} + s_{(u,v)} + s_{(v,d^1)} \end{aligned} \qquad \forall\ (u, v) \in A(W^*). \tag{5.77}$$

Posons $\lambda^1_{o^1} = 0$ et $\lambda^1_v = -r^1_{(o^1,v)}$ pour tout sommet $v \in V \setminus \{o^1\}$. Il s'ensuit que $r^1_{(o^1,v)} = \lambda^1_{o^1} - \lambda^1_v$ pour tout arc $(o^1, v) \in \delta^{\text{out}}(o^1)$. Soit (u, v) un arc de $A(\overline{W}^*) \cap A^1$ tel que $u \neq o^1$. D'après l'équation (5.75) et les valeurs r^1 associées aux arcs sortant de o^1, on obtient

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in A(\overline{W}^*) \cap A^1. \tag{5.78}$$

Soit (u, v) un arc sortant de W^* appartenant à A^1. L'équation (5.76) associée à l'arc (o^1, v) et au sommet u implique, d'après les valeurs de r^1 associées aux arcs sortant de o^1 et d'après les équations (5.67) et (5.78), que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v + \rho$. Comme (u, v) est choisi arbitrairement, on déduit que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v + \rho \qquad \forall\ (u, v) \in \delta^{\text{out}}(W^*) \cap A^1. \tag{5.79}$$

Considérons maintenant un arc (u, v) de $\delta^{\text{in}}(W^*)$ tel que $o^1 \neq u \neq d^1$. L'équation (5.76) associée à l'arc (u, d^1) et au sommet v implique que l'on a $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. D'après les valeurs associées aux arcs sortant de o^1, comme (u, v) est un arc quelconque de $\delta^{\text{in}}(W^*)$ tel que $o^1 \neq u \neq d^1$, il en résulte que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in \delta^{\text{in}}(W^*) \cap A^1. \tag{5.80}$$

Finalement, considérons un arc (u, v) appartenant à $A(W^*)$. D'après l'équation (5.77) associée à l'arc (u, v), on déduit d'après les valeurs de s et r^1 déjà données, que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. Comme (u, v) est choisi arbitrairement, il s'ensuit que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in A(W^*). \tag{5.81}$$

Comme la demande 1 est une demande quelconque de $A_\Phi(\overline{W}^*)$, on peut réitérer la preuve pour chaque demande de cet ensemble. On obtient alors

$$r^k_{(u,v)} = \begin{cases} \lambda^k_u - \lambda^k_v + \rho & \text{si } (u, v) \in \delta^{\text{out}}(W^*), \\ \lambda^k_u - \lambda^k_v & \text{sinon,} \end{cases} \qquad \forall\ k \in A_\Phi(\overline{W}^*), \forall\ (u, v) \in A^k \tag{5.82}$$

Les équations (5.67), (5.74) et (5.82) permettent de conclure que F_{W^*} est une facette de $P_U(D, v_0, K)$ si $|\overline{W}^*| \geq 3$, ce qui termine la preuve. □

5.1.2.5 Contraintes de connexité des demandes

Dans cette partie, nous étudions la structure faciale des contraintes de connexité des demandes. Soient k^* une demande de K, W^* un sous-ensemble de sommets de V tel que o^{k^*} et d^{k^*} appartiennent à W^* et u^* un sommet de $\overline{W}^*$. Soit $F^{k^*}_{W^*}$ la face induite par la contrainte de connexité des demandes (4.14) associée à k^*, W^* et u^*, c'est-à-dire,

$$F^{k^*}_{W^*} = \{(x, y) \in P_U(D, v_0, K) : \sum_{a \in \delta^{\text{out}}(W^*) \cap A^{k^*}} x^{k^*}_a - \sum_{a \in \delta^{\text{out}}(u^*) \cap A^{k^*}} x^{k^*}_a = 0\}.$$

On donne dans le théorème suivant les conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

Théorème 5.12 *Les contraintes de connexité des demandes* (4.14) *définissent des facettes si et seulement si* $|\overline{W}^*| \geq 2$.

Preuve. Supposons tout d'abord que $|\overline{W}^*| = 1$. On a alors $\overline{W}^* = \{u^*\}$. Dans ce cas, l'inégalité (4.14) est équivalente à l'inégalité $\sum_{a\in\delta^{\text{in}}(u^*)\cap A^{k^*}} x_a^{k^*} - \sum_{a\in\delta^{\text{out}}(u^*)\cap A^{k^*}} x_a^{k^*} \geq 0$. Le membre de gauche de cette inégalité est le même que celui de l'équation de conservation de flot (4.12) associée à k^* et u^*. La face induite par l'inégalité (4.14) correspond alors au polytope $P_U(D, v_0, K)$ et n'est donc pas une face propre. En conséquence, elle ne peut définir une facette.

Dans le reste de la preuve, on suppose que $|\overline{W}^*| \geq 2$. Soit $b^T x + c^T y \geq \alpha$ l'inégalité (4.14) associée à la demande k^*, au sous-ensemble de sommets W^* et au sommet u^*. Soit $r^T x + s^T y \geq \beta$ une inégalité valide définissant une facette F de $P_U(D, v_0, K)$. Supposons que $F_{W^*}^{k^*} \subseteq F$. On montre qu'il existe alors $\rho \in \mathbb{R}$, $\lambda = (\lambda_v^k, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$.

On s'intéresse d'abord au vecteur s. La proposition 5.2 assure qu'il existe une solution réalisable, disons (L^1, T^1), telle que $T^1 = \delta(o^{k^*})$. De plus, il est facile de voir que le vecteur d'incidence de cette solution appartient à $F_{W^*}^{k^*}$. Soient u, v et w trois sommets distincts différents de o^{k^*}. Les vecteurs d'incidence des solutions réalisables $(L^1, \bar{T}^1)$ et $(L^1, \tilde{T}^1)$ avec $\bar{T}^1 = T^1 \cup \{(u, v), (v, u)\}$ et $\tilde{T}^1 = T^1 \cup \{(u, v), (v, w), (w, u)\}$ appartiennent également à $F_{W^*}^{k^*}$. On a alors $x^{L^1} + y^{T^1} = x^{L^1} + y^{\bar{T}^1}$, ce qui implique que $s_{(u,v)} + s_{(v,u)} = 0$. De même, comme $x^{L^1} + y^{T^1} = x^{L^1} + y^{\tilde{T}^1}$, on a $s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0$. Comme u, v et w sont des sommets quelconques de $V \setminus \{o^{k^*}\}$, il s'ensuit que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall u \neq v \in V \setminus \{o^{k^*}\}, \tag{5.83}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall u \neq v \neq w \neq u \in V \setminus \{o^{k^*}\}. \tag{5.84}$$

Le vecteur d'incidence de la solution (L^1, T^1) reste un point de $F_{W^*}^{k^*}$ si $T^1 = \delta(d^{k^*})$ ou $T^1 = \delta(u^*)$. On peut alors rependre le même raisonnement en remplaçant successivement le sommet o^{k^*} par les sommets d^{k^*} et u^*, ce qui donne quatre nouvelles équations similaires à (5.83) et (5.84). On généralise alors les équations (5.83) et (5.84) selon

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall u \neq v \in V, \tag{5.85}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall u \neq v \neq w \neq u \in V, \{u, v, w\} \neq \{o^{k^*}, d^{k^*}, u^*\}. \tag{5.86}$$

Posons $\mu_{v_0} = 0$ et $\mu_v = -s_{(v_0,v)}$ pour tout sommet v de V différent de v_0. On a alors $s_{(v_0,v)} = \mu_{v_0} - \mu_v$ pour tout $v \in V \setminus \{v_0\}$. Par les équations (5.85), on déduit que $s_{(v,v_0)} = \mu_v - \mu_{v_0}$ pour tout $v \in V \setminus \{v_0\}$. Considérons un arc (u, v) n'appartenant pas à $\delta(v_0)$. Comme $v_0 \notin \{o^{k^*}, d^{k^*}, u^*\}$, on peut considérer l'équation (5.86) associée à v_0, u et v. D'après les valeurs des arcs incidents a v_0, on déduit que $s_{(u,v)} = \mu_u - \mu_v$. Comme (u, v) est un arc quelconque n'appartenant pas à $\delta(v_0)$ et d'après les valeurs des arcs incidents à v_0, il en résulte que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall (u, v) \in A. \tag{5.87}$$

Considérons maintenant une demande de $K \setminus \{k^*\}$. Supposons, sans perte de généralité, que cette demande est la demande 1. Soit (u,v) un arc de $A^1 \setminus \delta(o^1)$. Supposons que $(u,v) \neq (o^{k^*}, d^{k^*})$. Notons par D^1 le graphe induit par l'ensemble d'arcs $A \setminus \{(o^{k^*}, d^{k^*}), (o^1,u), (u,v), (v,d^1), (o^1,v)\}$. Il est facile de voir que D^1 est $(n-4)$-arc connexe. Comme $n \geq 2p+1$ et $p \geq 2$, il s'ensuit que D^1 est $(p-1)$-arc connexe. Il existe donc $p-1$ arborescences disjointes couvrantes, ayant respectivement v_0 et o^k, $k \in K \setminus \{1, k^*\}$ comme racine. Dans toute arborescence ayant o^k comme racine, on peut déduire un chemin élémentaire de o^k à d^k que l'on note L^4_k. Notons par Γ_1 l'arborescence couvrante ayant v_0 comme racine. Les chemins de L^4_k, $k \in K \setminus \{1, k^*\}$ sont donc arc-disjoints et n'intersecte pas Γ_1. Considérons les chemins $L^4_1 = \{(o^1,u), (u,v), (v,d^1)\}$ et $L^4_{k^*} = \{(o^{k^*}, d^{k^*})\}$. $L^4 = \{L^4_1, L^4_2, \ldots, L^4_p\}$ correspond alors à un ensemble de $o^k d^k$-chemins élementaires arc-disjoints, $k \in K$. De plus, Γ_1 n'intersecte aucun chemin aucun chemin de L^4. Comme Γ_1 est une v_0-arborescence couvrante, il s'ensuit, d'après la proposition 5.1, que (L^4, A) est une solution réalisable. D'après la définition de $L^4_{k^*}$, on a $x^{k^*}(\delta^{\text{out}}(W^*) \cap A^{k^*})) = 0$, ce qui implique que (x^{L^4}, y^A) appartient à $F^{k^*}_{W^*}$. Comme l'arc (o^1, v) n'appartient à aucun chemin de L^4, le vecteur d'incidence de la solution $(\bar{L}^4, A)$, avec

$$\bar{L}^4_k = \begin{cases} \{(o^1,v), (v,d^1)\} & \text{si } k = 1, \\ L^4_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K,$$

appartient également à $F^{k^*}_{W^*}$. On a alors $x^{L^4} + y^A = x^{\bar{L}^4} + y^A$, ce qui implique que $r^1_{(o^1,v)} = r^1_{(o^1,u)} + r^1_{(u,v)}$. Comme (u,v) est un quelconque de $A^1 \setminus \delta(o^1)$, il en résulte que

$$r^1_{(o^1,v)} = r^1_{(o^1,u)} + r^1_{(u,v)} \qquad \forall (u,v) \in A^1 \setminus \delta(o^1). \tag{5.88}$$

L'équation (5.88) reste vraie si $(u,v) = (o^{k^*}, d^{k^*})$. En effet, notons par $\bar{A}$ l'ensemble défini par

$$\bar{A} = \begin{cases} \{(o^{k^*}, u^*), (u^*, d^1), (d^1, d^{k^*}), (o^1, o^{k^*}), (o^{k^*}, d^{k^*}), (d^{k^*}, d^1), (o^1, d^{k^*})\} & \text{si } u^* = o^1, \\ \{(o^{k^*}, u^*), (u^*, d^{k^*}), (o^1, o^{k^*}), (o^{k^*}, d^{k^*}), (d^{k^*}, d^1), (o^1, d^{k^*})\} & \text{sinon.} \end{cases}$$

Considérons que le graphe D^1 est le graphe induit par $A \setminus \bar{A}$ et modifions la définition de $L^4_{k^*}$ tel que

$$L^4_{k^*} = \begin{cases} \{(o^{k^*}, u^*), (u^*, d^1), (d^1, d^{k^*})\} & \text{si } u^* = o^1, \\ \{(o^{k^*}, u^*), (u^*, d^{k^*})\} & \text{sinon.} \end{cases}$$

En reprenant la preuve précédente avec les nouvelles définitions de D^1 et L^4, on obtient alors l'équation (5.88). Posons $\lambda^1_{o^1} = 0$ et $\lambda^1_v = -r^1_{(o^1,v)}$ pour tout $v \in V \setminus \{o^1\}$. On a alors $r^1_{(o^1,v)} = \lambda^1_{o^1} - \lambda^1_v$ pour tout $v \in V \setminus \{o^1\}$. Considérons un arc (u,v) de A^1 tel que

$u \neq o^1$. Par l'équation (5.88) associée à (u,v) et d'après les valeurs des arcs sortant de o^1, il s'ensuit que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. Comme (u,v) est choisi arbitrairement, on peut étendre ce résultat à tous les arcs de A^1. Finalement, on peut reitérer le raisonnement pour toute demande de K différente de k^*. Il en résulte que

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\ k \in K \setminus \{k^*\}, \forall\ (u,v) \in A^k. \tag{5.89}$$

Considérons maintenant la demande k^*. Soit u_1 un sommet de $\overline{W}^*$ différent de u^*. Posons $\lambda^{k^*}_{d^{k^*}} = 0$ et $\lambda^{k^*}_v = r^{k^*}_{(v,d^{k^*})}$ pour tout $v \in V \setminus \{d^{k^*}, u^*\}$. On a alors $r^{k^*}_{(v,d^{k^*})} = \lambda^{k^*}_v - \lambda^{k^*}_{d^{k^*}}$ pour tout $v \in V \setminus \{d^{k^*}, u^*\}$. Posons également $\lambda^{k^*}_{u^*} = r^{k^*}_{(u_1,d^{k^*})} - r^{k^*}_{(u_1,u^*)}$ et $\rho = r^{k^*}_{(o^{k^*},u^*)} + r^{k^*}_{(u_1,d^{k^*})} - r^{k^*}_{(u_1,u^*)} - r^{k^*}_{(o^{k^*},d^{k^*})}$. On déduit que $r^{k^*}_{(u_1,u^*)} = \lambda^{k^*}_{u_1} - \lambda^{k^*}_{u^*}$ et $r^{k^*}_{(o^{k^*},u^*)} = \lambda^{k^*}_{o^{k^*}} - \lambda^{k^*}_{u^*} + \rho$.

Soit (u,v) un arc de $A(W^*) \cap A^{k^*}$ tel que $v \neq d^{k^*}$. Le graphe induit par l'ensemble d'arcs $A \setminus \{(o^{k^*},u),(u,v),(v,d^{k^*}),(u,d^{k^*})\}$ est $(n-3)$-arc connexe. Comme $n \geq 2p+1$ et $p \geq 2$, alors il est p-arc connexe. D'après la proposition 5.5, il existe une solution (L^5, A) telle que $L^5_{k^*} = \{(o^{k^*},u),(u,v),(v,d^{k^*})\}$. Comme u et v sont des sommets de W^*, le vecteur d'incidence de cette solution appartient à $F^{k^*}_{W^*}$. De plus, proposition 5.5 implique qu'il existe une autre solution réalisable, disons $(\bar{L}^5, A)$, avec

$$\bar{L}^5_k = \begin{cases} \{(o^{k^*},u),(u,d^{k^*})\} & \text{si } k = k^*, \\ L^5_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

De plus, $(x^{\bar{L}^5}, y^A)$ appartient à $F^{k^*}_{W^*}$, ce qui implique que $x^{L^5} + y^A = x^{\bar{L}^5} + y^A$. On déduit que $r^{k^*}_{(u,d^{k^*})} = r^{k^*}_{(u,v)} + r^{k^*}_{(v,d^{k^*})}$. D'après les valeurs de r^{k^*} associées aux arcs (u,d^{k^*}) et (v,d^{k^*}), il s'ensuit que $r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v$. Comme (u,v) est choisi arbitrairement et d'après les valeurs de r^{k^*} associées aux arcs entrants de d^{k^*}, il en résulte que

$$r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v \qquad \forall\ (u,v) \in A(W^*) \cap A^{k^*}. \tag{5.90}$$

Soit (u,v) un arc de A^{k^*} tel que u est un sommet de $\overline{W}^* \setminus \{u^*\}$ et v est un sommet de $V \setminus \{u^*, u, o^{k^*}, d^{k^*}\}$. Le graphe induit par l'ensemble d'arcs $A \setminus \{(o^{k^*},u^*),(u^*,u),(u,v),(v,d^{k^*}),(u,d^{k^*})\}$ est $(n-3)$-arc connexe. Ceci implique que le graphe est p-arc connexe puisque $p \geq 2$ et $n \geq 2p+1$. La proposition 5.5 implique qu'il existe une solution réalisable, disons (L^6, A), telle que $L^6_{k^*} = \{(o^{k^*},u^*),(u^*,u),(u,v),(v,d^{k^*})\}$. Comme u est un sommet de $\overline{W}^*$, un seul arc de $L^6_{k^*}$ appartient à la coupe sortante de W. De plus, $L^6_{k^*}$ contient un arc sortant de u^*, ce qui implique que la contrainte de connexité des demandes associée à k^*, W^* et u^* est vérifiée à l'égalité par (x^{L^6}, y^A). Ce point

est donc un point de $F^{k^*}_{W^*}$. La proposition 5.5 implique qu'il existe une autre solution, disons $(\bar{L}^6, A)$, telle que

$$\bar{L}^6_k = \begin{cases} \{(o^{k^*}, u^*), (u^*, u), (u, d^{k^*})\} & \text{si } k = k^*, \\ L^6_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Il est clair que le vecteur d'incidence de $(\bar{L}^6, A)$ appartient à $F^{k^*}_{W^*}$. On a alors $x^{L^6} + y^A = x^{\bar{L}^6} + y^A$, ce qui implique que $r^{k^*}_{(u,v)} + r^{k^*}_{(v,d^{k^*})} = r^{k^*}_{(u,d^{k^*})}$. D'après les valeurs de r^{k^*} associées aux arcs (u, d^{k^*}) et (v, d^{k^*}), on déduit que $r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v$. Comme (u, v) est choisi arbitrairement, il en résulte que

$$r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v \qquad \forall\ (u, v) \in A^{k^*} \cap (\delta^{\text{in}}(W^*) \cup A(\overline{W}^*)) \setminus \delta(u^*). \tag{5.91}$$

Soit v un sommet de W^* différent de o^{k^*} et d^{k^*}. Considérons le graphe induit par l'ensemble d'arcs $\{(o^{k^*}, v), (v, u^*), (u^*, d^{k^*}), (o^{k^*}, u^*)\}$ est p-arc connexe. D'après la proposition 5.5, il existe une solution, disons (L^7, A), telle que $L^7_{k^*} = \{(o^{k^*}, v), (v, u^*), (u^*, d^{k^*})\}$. De plus, il est clair que le vecteur d'incidence appartient à $F^{k^*}_{W^*}$. De plus, par la proposition 5.5, on peut construire une autre solution, disons $(\bar{L}^7, A)$, telle que

$$\bar{L}^7_k = \begin{cases} \{(o^{k^*}, u^*), (u^*, d^{k^*})\} & \text{si } k = k^*, \\ L^7_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Le vecteur $(x^{\bar{L}^7}, y^A)$ appartient à $F^{k^*}_{W^*}$. On déduit alors que $r^{k^*}_{(o^{k^*},u^*)} = r^{k^*}_{(o^{k^*},v)} + r^{k^*}_{(v,u^*)}$. D'après les valeurs $r^{k^*}_{(o^{k^*},u^*)}$ et $r^{k^*}_{(o^{k^*},v)}$, il s'ensuit que $r^{k^*}_{(v,u^*)} = \lambda^{k^*}_v - \lambda^{k^*}_{u^*} + \rho$. Comme v est un sommet quelconque de $W^* \setminus \{o^{k^*}, d^{k^*}\}$ et d'après la valeur $r^{k^*}_{(o^{k^*},u^*)}$, il en résulte que

$$r^{k^*}_{(v,u^*)} = \lambda^{k^*}_v - \lambda^{k^*}_{u^*} + \rho \qquad \forall\ v \in W^* \setminus \{d^{k^*}\}. \tag{5.92}$$

Soit v un sommet de V différent de u^* et o^{k^*}. Comme le graphe induit par l'ensemble d'arcs $A \setminus \{(o^{k^*}, u^*), (u^*, v), (v, d^{k^*}), (o^{k^*}, d^{k^*})\}$ est $(n-3)$-arc connexe, il est p-arc connexe puisque $p \geq 2$ et $n \geq 2p + 1$. D'après la proposition 5.5, il existe deux solutions, disons (L^8, A) et $(\bar{L}^8, A)$, réalisables pour le 1-PRLPA unitaire telles que $L^8_k = \bar{L}^8_k$ pour tout $k \neq k^*$, $L^8_{k^*} = \{(o^{k^*}, u^*), (u^*, v), (v, d^{k^*})\}$ et $\bar{L}^8_{k^*} = \{(o^{k^*}, d^{k^*})\}$. De plus, il est facile de voir que ces deux solutions appartiennent à $F^{k^*}_{W^*}$. Comme les vecteurs d'incidence de ces deux solutions vérifient à l'égalité la contrainte $r^T c + s^T y \geq \beta$, on déduit que $r^{k^*}_{(o^{k^*},u^*)} + r^{k^*}_{(u^*,v)} + r^{k^*}_{(v,d^{k^*})} = r^{k^*}_{(o^{k^*},d^{k^*})}$. D'après les valeurs de r^{k^*} associées aux arcs (o^{k^*}, u^*), (v, d^{k^*}) et (o^{k^*}, d^{k^*}), on obtient $r^{k^*}_{(u^*,v)} = \lambda^{k^*}_{u^*} - \lambda^{k^*}_v - \rho$. En appliquant le même raisonnement pour tout sommet de $V \setminus \{u^*, o^{k^*}\}$, on obtient

$$r^{k^*}_{(u^*,v)} = \lambda^{k^*}_{u^*} - \lambda^{k^*}_v - \rho \qquad \forall\ (u^*, v) \in A^{k^*}. \tag{5.93}$$

Supposons que $|\overline{W}^*| \geq 3$. Soit v un sommet de $\overline{W}^*$ différent de u^* et u_1. De la même manière que précédemment, on montre qu'il existe deux solutions, disons (L^9, A) et $(\bar{L}^9, A)$, appartenant à $F^{k^*}_{W^*}$ telles que $L^9_k = \bar{L}^9_k$ pour tout $k \neq k^*$, $L^9_{k^*} = \{(o^{k^*}, u_1), (u_1, v), (v, u^*), (u^*, d^{k^*})\}$ et $\hat{L}^9_{k^*} = \{(o^{k^*}, u_1), (u_1, u^*), (u^*, d^{k^*})\}$. On obtient alors $r^{k^*}_{(u_1,v)} + r^{k^*}_{(v,u^*)} = r^{k^*}_{(u_1,u^*)}$, ce qui implique, d'après les valeurs de r^{k^*} déjà données, que $r^{k^*}_{(v,u^*)} = \lambda^{k^*}_v - \lambda^{k^*}_{u^*}$. En appliquant le même raisonnement pour tout sommet de $\overline{W}^* \setminus \{u^*\}$, on déduit que

$$r^{k^*}_{(u,u^*)} = \lambda^{k^*}_u - \lambda^{k^*}_{u^*} \qquad \forall\ (u, u^*) \in A(\overline{W}^*) \cap A^{k^*}. \tag{5.94}$$

Finalement, soit (u, v) un arc de $\delta^{\text{out}}(W^*) \cap A^{k^*}$, avec $v \neq u^*$. Comme le graphe induit par l'ensemble d'arcs $A \setminus \{(o^{k^*}, u), (u, v), (v, u^*), (u^*, d^{k^*}), (o^{k^*}, d^{k^*})\}$ est p-arc connexe, on sait qu'il existe, d'après la proposition 5.5, deux solutions, disons (L^{10}, A) et $(\bar{L}^{10}, A)$, réalisables pour le 1-PRLPA unitaire telles que $L^{10}_k = \bar{L}^{10}_k$ pour tout $k \neq k^*$, $L^{10}_{k^*} = \{(o^{k^*}, u), (u, v), (v, u^*), (u^*, d^{k^*})\}$ et $\bar{L}^{10}_{k^*} = \{(o^{k^*}, d^{k^*})\}$. De plus, ces deux solutions appartiennent à $F^{k^*}_{W^*}$. On déduit alors que $r^{k^*}_{(o^{k^*},u)} + r^{k^*}_{(u,v)} + r^{k^*}_{(v,u^*)} + r^{k^*}_{(u^*,d^{k^*})} = r^{k^*}_{(o^{k^*},d^{k^*})}$. Comme les valeurs du vecteur $r^{k^*}_a$ sont fixées pour tout arc $a \in A^{k^*} \setminus \delta^{\text{out}}(W^*)$ et pour tout arc $a \in A^{k^*} \cap \delta^{\text{in}}(u^*)$, on déduit que $r^{k^*}_{(u,v)} = \lambda^{k^*}_u - \lambda^{k^*}_v + \rho$. En réitérant pour tout arc sortant de W^* appartenant à A^{k^*}, on obtient, à l'aide des équations (5.90)-(5.94), que

$$r^{k^*}_{(u,v)} = \begin{cases} \lambda^{k^*}_u - \lambda^{k^*}_v + \rho & \text{si } (u, v) \in \delta^{\text{out}}(W^*), \\ \lambda^{k^*}_u - \lambda^{k^*}_v - \rho & \text{si } (u, v) \in \delta^{\text{out}}(u^*), \\ \lambda^{k^*}_u - \lambda^{k^*}_v & \text{sinon,} \end{cases} \qquad \forall\ (u, v) \in A^{k^*}. \tag{5.95}$$

Les équations (5.87), (5.89) et (5.95) permettent de conclure que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$. La face $F^{k^*}_{W^*}$ définit donc une facette de $P_U(D, v_0, K)$. □

Nous étudions maintenant la structure faciale des contraintes de connexité (4.10).

5.1.2.6 Contraintes de connexité

Dans cette partie, nous étudions la structure faciale des contraintes de connexité (4.10). Soient W^* un sous-ensemble de sommets de V, avec $v_0 \in W^*$, et (u^*, v^*) un arc de $A(\overline{W}^*)$. Soit $F^{(u^*,v^*)}_{W^*}$ la face définie par la contrainte de connexité (4.10) associée à W^* et (u^*, v^*), *i.e.*,

$$F^{(u^*,v^*)}_{W^*} = \{(x, y) \in P_U(D, v_0, K) : \sum_{a \in \delta^{\text{out}}(W^*)} y_a - y_{(u^*,v^*)} = 0\}.$$

On donne dans le théorème suivant des conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

Théorème 5.13 *Les inégalités de connexité* (4.10) *définissent des facettes pour le polytope* $P_U(D, v_0, K)$ *si et seulement si* $\delta_\Phi(W^*) = \emptyset$, $A_\Phi(\overline{W}^*) = \emptyset$ *et* $|\overline{W}^*| \geq 3$.

Preuve. Il est évident que si $\delta_\Phi(W^*) \neq \emptyset$, alors $\sum_{k\in K} x^k(\delta(W^*) \cap A^k) \geq 1$. Par les contraintes de capacité (4.15), on a $y(\delta^{\text{out}}(W^*)) \geq 1$ ou $y(\delta^{\text{in}}(W^*)) \geq 1$. Dans les deux cas, les contraintes de conservation de flot impliquent que $y(\delta^{\text{out}}(W^*)) \geq 1$. Comme $y_a \leq 1$ pour tout arc $a \in A$, cette dernière inégalité domine la contrainte (4.10). De plus, la face induite par l'inégalité est différente de $P_U(D, v_0, K)$. La contrainte (4.10) ne définit donc pas une facette. On suppose, pour le reste de la preuve, que $\delta_\Phi(W^*) = \emptyset$.

Supposons que $A_\Phi(\overline{W}^*) \neq \emptyset$. Dans ce cas, l'inégalité (4.10) est clairement dominée par la contrainte de vulnérabilité renforcée (4.24) associée à W^*. Comme la face induite par cette dernière ne correspond pas au polytope $P_U(D, v_0, K)$, la contrainte de connexité ne définit pas une facette. On suppose, pour le reste de la preuve, que $A_\Phi(\overline{W}^*) = \emptyset$.

Supposons maintenant que $|\overline{W}^*| = 2$, *i.e.*, $\overline{W}^* = \{u^*, v^*\}$. Supposons qu'il existe une demande, disons la demande 1, passant par l'arc (v^*, u^*). Comme o^1 et d^1 appartiennent à W^*, il existe deux sommets de W^*, disons v_1 et v_2, avec (v_1, v^*) et (u^*, v_2) appartenant au chemin de la demande 1. Les contraintes de capacité impliquent que $y_{(v_1,v^*)} = 1$ et $y_{(u^*,v_2)} = 1$. Si $y_{(u^*,v^*)} = 0$, alors on a $y(\delta^{\text{out}}(W^*)) \geq 1$. Sinon, comme y vérifie les contraintes de conservation de flot et $y_{(u^*,v_2)} = y_{(u^*,v^*)} = 1$, il existe alors un sommet $v \in W^*$ tel que $y_{(v,u^*)} = 1$. On a alors $y(\delta^{\text{out}}(W^*)) \geq 2$. La contrainte de connexité est donc dominée par la contrainte valide $y(\delta^{\text{out}}(W^*)) - y_{(u^*,v^*)} - x_{(v^*,u^*)}(K^{(v^*,u^*)}) \geq 0$. Comme la face induite par cette inégalité est différente de $P_U(D, v_0, K)$, la contrainte (4.10) ne définit pas une facette. On suppose, pour le reste de la preuve, que $|\overline{W}^*| \geq 3$.

On remarque que $|W^*| \geq 5$ car on a $p \geq 2$, $v_0 \in W^*$ et $o^k, d^k \in W^*$ pour toute demande $k \in K$. Soit $b^T x + c^T y \geq \alpha$ l'inégalité (4.10) associée au sous-ensemble W^* et à l'arc (u^*, v^*). Soit $r^T x + s^T y \geq \beta$ une inégalité valide définissant une facette F de $P_U(D, v_0, K)$. Supposons que $F_{W^*}^{(u^*,v^*)} \subseteq F$. On montre qu'il existe alors $\rho \in \mathbb{R}$, $\lambda = (\lambda_v^k, v \in V, k \in K) \in \mathbb{R}^{np}$ et $\mu = (\mu_v, v \in V) \in \mathbb{R}^n$ tels que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$.

Soit $\bar{v}$ un sommet de W^*. Considérons la solution (L^1, T^1) définie par $L_k^1 = \{(o^k, \bar{v}), (\bar{v}, d^k)\}$ pour tout $k \in K$ et $T^1 = \delta(\bar{v}) \cap A(W^*)$. Il est facile de voir que (L^1, T^1) est réalisable. En effet, L^1 est un ensemble de $o^k d^k$-chemins élémentaires arc-disjoints, $k \in K$. De plus, l'ensemble $\Gamma_1 = \{(\bar{v}, o^k) : k \in K\} \cup \{(v_0, \bar{v})\}$ définit une v_0-arborescence couvrant les origines de K et n'intersectant aucun chemin de L^1. La proposition 5.1

assure que (L^1, T^1) est réalisable. De plus, comme T^1 ne contient aucun arc sortant de W^*, le vecteur d'incidence (x^{L^1}, y^{T^1}) appartient à $F_{W^*}^{(u^*,v^*)}$. Soient u, v et w trois sommets distincts de $W^* \setminus \{\bar{v}\}$. Les vecteurs d'incidence des solutions réalisables $(L^1, \bar{T}^1)$ et $(L^1, \tilde{T}^1)$ avec $\bar{T}^1 = T^1 \cup \{(u,v),(v,u)\}$ et $\tilde{T}^1 = T^1 \cup \{(u,v),(v,w),(w,u)\}$ appartiennent également à $F_{W^*}^{(u^*,v^*)}$. On a alors $x^{L^1} + y^{T^1} = x^{L^1} + y^{\bar{T}^1}$, ce qui implique que $s_{(u,v)} + s_{(v,u)} = 0$. De même, comme $x^{L^1} + y^{T^1} = x^{L^1} + y^{\tilde{T}^1}$, on a $s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0$. Comme u, v et w sont des sommets quelconques de $W^* \setminus \{\bar{v}\}$, il s'ensuit que

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall u \neq v \in W^* \setminus \{\bar{v}\}, \tag{5.96}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall u \neq v \neq w \neq u \in W^* \setminus \{\bar{v}\}. \tag{5.97}$$

Le vecteur d'incidence de la solution (L^1, T^1) reste un point de $F_{W^*}^{(u^*,v^*)}$ si le sommet $\bar{v}$ est remplacé par un sommet quelconque de $W^* \setminus \{\bar{v}\}$. Comme $|W^*| \geq 5$, on peut alors rependre le même raisonnement en remplaçant successivement le sommet $\bar{v}$ par trois autres sommets distincts de $W^* \setminus \{\bar{v}\}$, ce qui donne six nouvelles équations similaires à (5.96) et (5.97). On généralise alors les équations (5.96) et (5.97) selon

$$s_{(u,v)} + s_{(v,u)} = 0 \qquad \forall u \neq v \in W^*, \tag{5.98}$$

$$s_{(u,v)} + s_{(v,w)} + s_{(w,u)} = 0 \qquad \forall u \neq v \neq w \neq u \in W^*. \tag{5.99}$$

Considérons maintenant la solution (L^2, T^2) avec $L^2_k = \{(o^k, d^k)\}$ pour tout $k \in K$ et $T^2 = A(W^*) \cup \{(v_0, u^*), (u^*, v^*), (v^*, v_0)\}$. Il est clair que (L^2, T^2) est réalisable. De plus, comme T^2 contient un arc sortant de W^* et l'arc (u^*, v^*), alors la contrainte de connexité associée à W^* et (u^*, v^*) est vérifiée à l'égalité par (x^{L^2}, y^{T^2}). (x^{L^2}, y^{T^2}) appartient donc à $F_{W^*}^{(u^*,v^*)}$. Soit v un sommet de $\overline{W}^* \setminus \{u^*, v^*\}$. Les vecteurs d'incidence des solutions $(L^2, \bar{T}^2)$ et $(L^2, \tilde{T}^2)$ avec $\bar{T}^2 = T^2 \cup \{(v^*, v), (v, v^*)\}$ et $\tilde{T}^2 = T^2 \cup \{(u^*, v), (v, u^*)\}$ appartiennent également à $F_{W^*}^{(u^*,v^*)}$, ce qui implique que $s_{(v^*,v)} + s_{(v,v^*)} = 0$ et $s_{(u^*,v)} + s_{(v,u^*)} = 0$. Comme v est un sommet quelconque de $\overline{W}^* \setminus \{u^*, v^*\}$, on obtient alors

$$s_{(u^*,v)} + s_{(v,u^*)} = 0 \qquad \forall\ v \in \overline{W}^* \setminus \{u^*, v^*\}, \tag{5.100}$$

$$s_{(v^*,v)} + s_{(v,v^*)} = 0 \qquad \forall\ v \in \overline{W}^* \setminus \{u^*, v^*\}. \tag{5.101}$$

Considérons maintenant un sommet $u \in \overline{W}^* \setminus \{v, v^*\}$. Le vecteur d'incidence de la solution $(L^2, \hat{T}^2)$ avec $\hat{T}^2 = T^2 \cup \{(v^*, u), (u, v), (v, v^*)\}$ appartient également à $F_{W^*}^{(u^*,v^*)}$. Comme $x^{L^2} + y^{T^2} = x^{L^2} + y^{\hat{T}^2}$, il s'ensuit que $s_{(v^*,u)} + s_{(u,v)} + s_{(v,v^*)} = 0$. Comme u et v sont choisis de manière arbitraire, on peut étendre ce résultat pour obtenir

$$s_{(v^*,u)} + s_{(u,v)} + s_{(v,v^*)} = 0 \qquad \forall\ u \neq v \in \overline{W}^* \setminus \{v^*\}, v \neq u^*. \tag{5.102}$$

On s'intéresse maintenant aux arcs de la coupe $\delta(W^*)$. Reconsidérons le sommet $\bar{v}$ de W^* et la solution (L^1, T^1) définis précédemment. Soient u et w deux sommets

distincts de $W^* \setminus \{\bar{v}\}$ et v un sommet de $\overline{W}^*$. Considérons les solutions (L^1, T^3), $(L^1, \bar{T}^3)$ et $(L^1, \tilde{T}^3)$, avec $T^3 = T^1 \cup A(\overline{W}^*) \cup \{(u,v),(v,u)\}$, $\bar{T}^3 = T^1 \cup A(\overline{W}^*) \cup \{(u,v),(v,w),(w,u)\}$ et $\tilde{T}^3 = T^1 \cup A(\overline{W}^*) \cup \{(u,w),(w,v),(v,u)\}$. Comme (L^1, T^1) est réalisable, ces trois solutions le sont aussi. De plus, T^1 ne contient aucun arc sortant de W^*. Les ensembles T^3, $\bar{T}^3$ et $\tilde{T}^3$ contiennent donc exactement un arc sortant de W^* ainsi que l'arc (u^*, v^*). Il s'ensuit que les vecteurs d'incidence de ces trois solutions appartiennent à $F_{W^*}^{(u^*,v^*)}$. Comme $x^{L^1} + y^{T^3} = x^{L^1} + y^{\bar{T}^3}$, on déduit que $s_{(v,u)} = s_{(v,w)} + s_{(w,u)}$. Comme $x^{L^1} + y^{T^3} = x^{L^1} + y^{\tilde{T}^3}$, $s_{(u,v)} = s_{(u,w)} + s_{(w,v)}$. En réitérant ce raisonnement pour tous les sommets de $W^* \setminus \{\bar{v}\}$ et de $\overline{W}^*$, on obtient alors

$$s_{(u,v)} = s_{(u,w)} + s_{(w,v)} \qquad \forall\ (u,v) \in \delta(W^*), w \in W^* \setminus \{u,v\}. \tag{5.103}$$

Par ailleurs, à l'aide d'un raisonnement similaire, on peut étendre ce résultat si w est un sommet de $\overline{W^*}$ de la manière suivante

$$s_{(u,v)} = s_{(u,w)} + s_{(w,v)} \qquad \begin{array}{l} \forall\ (u,v) \in \delta(W^*), w \in \overline{W}^* \setminus \{u,v\}, \\ (u,w) \neq (u^*,v^*) \neq (w,v). \end{array} \tag{5.104}$$

Finalement, la solution $(L^1, \hat{T}^3)$, avec $\hat{T}^3 = T^1 \cup \{(v_0,u^*),(u^*,v^*),(v^*,v_0)\}$ est également réalisable et $(x^{L^1}, y^{\hat{T}^3})$ est un point de $F_{W^*}^{(u^*,v^*)}$. Il s'ensuit que

$$s_{(v_0,u^*)} + s_{(u^*,v^*)} + s_{(v^*,v_0)} = 0. \tag{5.105}$$

Posons $\mu_{v_0} = 0$ et $\mu_v = -s_{(v_0,v)}$ pour tout $v \in W^* \setminus \{v_0\}$. On a alors $s_{(v_0,v)} = \mu_{v_0} - \mu_v$ pour tout $v \in W^* \setminus \{v_0\}$. Par les équations (5.98), il s'ensuit que $s_{(v,v_0)} = \mu_v - \mu_{v_0}$ pour tout $v \in W^* \setminus \{v_0\}$. Considérons un arc (u,v) de $A(W^*)$ tel que u et v soient différents de v_0. Par l'équation (5.99) associée à u, v et v_0, on déduit, d'après la valeur de s associées aux arcs incidents à v_0, que $s_{(u,v)} = \mu_u - \mu_v$. Comme (u,v) est un arc quelconque de $A(W^*)$ tel que u et v soient différents de v_0, on déduit que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u,v) \in A(W^*). \tag{5.106}$$

Posons $\mu_{v^*} = s_{(v^*,v_0)}$ et $\mu_v = s_{(v^*,v_0)} - s_{(v^*,v)}$ pour tout $v \in \overline{W}^* \setminus \{v^*\}$. On a alors $s_{(v^*,v_0)} = \mu_{v^*} - \mu_{v_0}$ et $s_{(v^*,v)} = \mu_{v^*} - \mu_v$ pour tout $v \in \overline{W}^* \setminus \{v^*\}$. Par les équations (5.101), il s'ensuit que $s_{(v,v^*)} = \mu_v - \mu_{v^*}$ pour tout $v \in W^* \setminus \{u^*, v^*\}$. Considérons un arc (u,v) de $A(\overline{W}^*) \setminus \delta(v^*)$. Supposons d'abord que (u,v) n'appartient pas à $\delta^{\text{in}}(u^*)$. Dans ce cas, par l'équation (5.102) et les valeurs de s associées aux arcs incidents à v^*, on déduit que $s_{(u,v)} = \mu_u - \mu_v$. Supposons maintenant que $v = u^*$. Dans ce cas, en utilisant le raisonnement précédent, on montre que $s_{(u^*,u)} = \mu_{u^*} - \mu_u$. D'après les

équations (5.100), on a $s_{(u^*,u)} + s_{(u,u^*)} = 0$, ce qui implique que $s_{(u,u^*)} = \mu_u - \mu_{u^*}$. En répétant ce raisonnement pour tout arc de $A(\overline{W}^*) \setminus \delta(v^*)$ et d'après les valeurs de s associées aux arcs incidents à v^*, il en résulte que

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u,v) \in A(\overline{W}^*) \setminus \{(u^*, v^*)\}. \tag{5.107}$$

Considérons un sommet v de $W^* \setminus \{v_0\}$. D'après l'équation (5.103) associée à l'arc (v^*, v) et au sommet v_0, on déduit que $s_{(v^*,v)} = \mu_{v^*} - \mu_v$. De plus, on peut étendre ce résultat à tous les arcs sortant de v^* et appartenant à $\delta^{\text{in}}(W^*)$. Considérons maintenant un arc $(u,v) \in \delta^{\text{in}}(W^*)$ tel que $u \neq v^*$. L'équation (5.104) associée à l'arc (v^*, v) et au sommet u implique que l'on a $s_{(u,v)} = \mu_u - \mu_v$. D'après la définition de (u,v) et les valeurs de s associées aux arcs sortant de v^*, on obtient

$$s_{(u,v)} = \mu_u - \mu_v \qquad \forall\ (u,v) \in \delta^{\text{in}}(W^*). \tag{5.108}$$

Posons $\rho = s_{(v^*,v_0)} + s_{(v^*,v_0)}$. Ceci implique que $s_{(v_0,v^*)} = \mu_{v_0} - \mu_{v^*} + \rho$. Soit v un sommet de $\overline{W} \setminus \{v^*\}$. D'après (5.104) associées à l'arc (v_0, v) et au sommet v^*, on obtient, d'après les valeurs de s déjà données, $s_{(v_0,v)} = \mu_{v_0} - \mu_v + \rho$. Considérons maintenant un arc $(u,v) \in \delta^{\text{out}}(W^*)$, avec $u \neq v_0$. L'équation (5.103) associée à l'arc (u,v) et au sommet v_0 implique que l'on a $s_{(u,v)} = \mu_u - \mu_v + \rho$. En répétant ce raisonnement pour tout arc sortant de W^*, on déduit que

$$s_{(u,v)} = \mu_u - \mu_v + \rho \qquad \forall\ (u,v) \in \delta^{\text{out}}(W^*). \tag{5.109}$$

Finalement, l'équation (5.105) assure que $s_{(v_0,u^*)} + s_{(u^*,v^*)} + s_{(v^*,v_0)} = 0$. D'après les valeurs de s associées à (v_0, u^*) et (v^*, v_0), il en résulte que $s_{(u^*,v^*)} = \mu_{u^*} - \mu_{v^*} - \rho$. D'après les équations (5.106)-(5.109), on obtient

$$s_{(u,v)} = \begin{cases} \mu_u - \mu_v + \rho & \text{si } (u,v) \in \delta^{\text{out}}(W^*), \\ \mu_u - \mu_v - \rho & \text{si } (u,v) = (u^*, v^*), \\ \mu_u - \mu_v & \text{sinon,} \end{cases} \qquad \forall\ (u,v) \in A. \tag{5.110}$$

Considérons une demande de K, par exemple la demande 1. Posons $\lambda^1_{o^1} = 0$ et $\lambda^1_v = -r^1_{(o^1,v)}$ pour tout $v \in V \setminus \{o^1\}$. On a alors $r^1_{(o^1,v)} = \lambda^1_{o^1} - \lambda^1_v$ pour tout $v \in V \setminus \{o^1\}$. On s'intéresse aux valeurs de r^1 associées aux arcs de $A(W^*) \cap A^1$. Soit (u,v) un arc de cet ensemble n'appartenant pas à $\delta(o^1)$. Soit $B_1 = A(W^*) \setminus \{(o^1,u), (u,v), (v,d^1), (o^1,v)\}$. Notons par n' la cardinalité de W^*. Comme $n' \geq 5$, le graphe induit par B_1 est $(n'-3)$-arc connexe, ce qui implique qu'il est p-arc connexe, puisque $n' \geq 2p+1$. D'après la proposition 5.5, il existe deux solutions réalisables, disons $(L^4, A(W^*))$ et $(\bar{L}^4, A(W^*))$,

telles que $L_1^4 = \{(o^1,u),(u,v),(v,d^1)\}$, $\bar{L}_1^4 = \{(o^1,v),(v,d^1)\}$ et $L_k^4 = \bar{L}_k^4$ pour tout $k \neq 1$. On déduit alors que $r^1_{(o^1,u)} + r^1_{(u,v)} = r^1_{(o^1,v)}$. D'après les valeurs de r^1 associées à (o^1,u) et (o^1,v), il s'ensuit que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. Comme (u,v) est choisi arbitrairement, on déduit que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u,v) \in A(W^*) \cap A^1. \tag{5.111}$$

On s'intéresse maintenant aux valeurs de r^1 associées aux arcs de $\delta(W^*) \cap A^1$. Soit $u \in W^* \setminus \{d^1\}$. On suppose que u est différent de v_0 ; la preuve pour le cas $u = v_0$ étant similaire, elle n'est pas donnée. Considérons l'ensemble $B_2 = A(W^*) \setminus \{(d^1,u),(u,v_0),(v_0,d^1)\}$. Le graphe induit par $B_2 \setminus \{(u,d^1),(o^1,u)\}$ est p-arc connexe. Il existe donc p arborescence disjointes ayant respectivement comme sommet d'origine v_0 ou o^k, $k \in K \setminus \{1\}$. Pour chaque o^k-arborescence, $k \neq 1$, on note par L_k^5 le chemin élémentaire de o^k à d^k. Notons par Γ_1 la v_0-arborescence. Il est clair que Γ_1 couvre toutes les origines des demandes de K. Soient v un sommet de $\overline{W}^*$ et $T^5 = B_2 \cup A(\overline{W}^*) \cup \{(u,v),(v,d^1),(d^1,u)\}$. Le graphe induit par T^5 est Eulérien. Notons par L_1^5 le chemin $\{(o^1,u),(u,v),(v,d^1)\}$. L'ensemble $L^5 = \{L_1^5, L_2^5, \ldots, L_p^5\}$ correspond alors à un ensemble de $o^k d^k$-chemins élémentaires arc-disjoints, $k \in K$, du graphe induit par T^5. De plus, aucun chemin de L^5 n'intersecte Γ_1. Comme Γ_1 couvre toutes les origines de K, la solution (L^5, T^5) est réalisable par la proposition 5.1. De plus, comme T^5 ne contient qu'un seul arc sortant de W^* et l'arc (u^*,v^*), (x^{L^5}, y^{T^5}) appartient à $F_{W^*}^{(u^*,v^*)}$. Par ailleurs, considérons l'ensemble de chemins $\bar{L}^5$ défini par

$$L_k^5 = \begin{cases} \{(o^1,u),(u,d^1)\} & \text{si } k = 1, \\ L_k^5 & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Il est facile de voir, en utilisant la même preuve que pour (x^{L^5}, y^{T^5}), que $(x^{\bar{L}^5}, y^{T^5})$ appartient à $F_{W^*}^{(u^*,v^*)}$. On a alors $x^{L^5} + y^{T^5} = x^{\bar{L}^5} + y^{T^5}$, ce qui implique que l'on a $r^1_{(u,d^1)} = r^1_{(u,v)} + r^1_{(v,d^1)}$. On déduit alors que

$$r^1_{(u,d^1)} = r^1_{(u,v)} + r^1_{(v,d^1)} \qquad \forall\ (u,v) \in \delta^{\text{out}}(W^*) \cap A^1. \tag{5.112}$$

En utilisant un raisonnement similaire, on peut montrer que r^1 vérifie

$$r^1_{(o^1,v)} = r^1_{(o^1,u)} + r^1_{(u,v)} \qquad \forall\ (u,v) \in \delta^{\text{in}}(W^*) \cap A^1. \tag{5.113}$$

Considérons l'équation (5.112) associée à un arc (o^1,v), où $v \in \overline{W}^*$. On obtient alors que $r^1_{(o^1,d^1)} = r^1_{(o^1,v)} + r^1_{(v,d^1)}$, ce qui implique que $r^1_{(v,d^1)} = \lambda^1_v - \lambda^1_{d^1}$ pour tout $v \in \overline{W}^*$. Considérons maintenant un arc (u,v) appartenant à $\delta^{\text{out}}(W^*) \cap A^1$. Par (5.112) associée à (u,v), on déduit, d'après les valeurs de r^1 associées aux arcs entrants de d^1, que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. Finalement, considérons un arc (u,v) de $\delta^{\text{in}}(W^*) \cap A^1$. Par l'équation

(5.113) associée à (u, v) et d'après les valeurs des arcs sortant de W^* et appartenant à A^1, il s'ensuit que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. Il en résulte que

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in \delta(W^*) \cap A^1. \tag{5.114}$$

On s'intéresse maintenant aux valeurs de r^1 associées aux arcs de $A(\overline{W}^*)$. Soit (u, v) un arc de cet ensemble. Comme $|\overline{W}^*| \geq 3$, il existe $w \in \overline{W}^*$ différent de u et v. Posons

$$\bar{A} = \begin{cases} \{(v, u)\} & \text{si } (v, u) \neq (u^*, v^*), \\ \{(v, w), (w, u)\} & \text{sinon.} \end{cases}$$

Ajoutons à $\bar{A}$ les arcs (o^1, v_0) et (v_0, d^1). Soit (L^6, T^6) une solution de $F^{(u^*,v^*)}_{W^*}$ avec $T^6 = A(W^*) \cup A(\overline{W}^*) \cup \{(o^1, u), (v, d^1)\} \setminus \bar{A}$ et $L^6_k = \{(o^k, d^k)\}$ pour tout $k \in K$. Il est facile de voir que (L^6, T^6) est réalisable. De plus, comme T^6 contient un seul arc sortant de W^* et l'arc (u^*, v^*), alors le vecteur d'incidence de (L^6, T^6) appartient à $F^{(u^*,v^*)}_{W^*}$. considérons la solution $(\hat{L}^6, T^6)$ définie par

$$\bar{L}^6_k = \begin{cases} \{(o^1, u), (u, v), (v, d^1)\} & \text{si } k = 1, \\ L^6_k & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

$(x^{\bar{L}^6}, y^{T^6})$ appartient également à $F^{(u^*,v^*)}_{W^*}$. Ceci implique que $r^1_{(o^1,d^1)} = r^1_{(o^1,u)} + r^1_{(u,v)} + r^1_{(v,d^1)}$. D'après les équations (5.111) et (5.114), on déduit que $r^1_{(u,v)} = \lambda^1_u - \lambda^1_v$. En appliquant ce raisonnement pour tout arc de $A(\overline{W}^*)$, on obtient

$$r^1_{(u,v)} = \lambda^1_u - \lambda^1_v \qquad \forall\ (u, v) \in A(\overline{W}^*). \tag{5.115}$$

Comme la demande 1 a été choisie arbitrairement, on peut réitérer la preuve pour n'importe quelle demande de K. On obtient alors

$$r^k_{(u,v)} = \lambda^k_u - \lambda^k_v \qquad \forall\ k \in K, (u, v) \in A^k. \tag{5.116}$$

Les équations (5.110) et (5.116) impliquent que $(r, s) = \rho(b, c) + (\lambda, \mu)M_{CF}$ si $\delta_\Phi(W^*) = \emptyset$, $|\overline{W}^*| \geq 3$ et $A_\Phi(\overline{W}^*) = \emptyset$, ce qui termine la preuve. □

On s'intéresse, dans la section suivante, à la résolution exacte du problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire à l'aide d'un algorithme de coupes et branchements.

5.2 Algorithme de coupes et branchements

Dans cette section, nous présentons un algorithme de coupes et branchements pour résoudre le problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire. Cet algorithme est basé sur les résultats polyédraux obtenus dans la section précédente. Les résultats expérimentaux obtenus à l'aide de cet algorithme montrent l'efficacité de la formulation introduite pour ce problème dans le chapitre 4.

5.2.1 Description de l'algorithme

L'algorithme calcule une solution optimale ou proche de l'optimum pour le 1-PRLPA unitaire formulé par le programme linéaire en nombres entiers (4.23). Une instance est définie par un triplet (D, v_0, K) et un vecteur coût $c \in \mathbb{R}^A$ associé aux arcs du graphe D. Le coût c_a représente le coût engendré par le passage du véhicule sur cet arc. Dans cet algorithme, on suppose que le graphe D est complet, *i.e.*, pour tout couple de sommets $u \neq v$ de V, l'arc (u, v) appartient au graphe. On suppose également que chaque sommet est incident à au plus une demande et que le dépôt n'est incident à aucune demande. Sous ces hypothèses, des conditions nécessaires et suffisantes ont été données pour que les contraintes (4.10), (4.14)-(4.20),(4.24) définissent des facettes, ce qui permet à l'algorithme de coupes et branchements de considérer uniquement des contraintes essentielles.
Le vecteur c est supposé strictement positif. En effet, dans ce type de problème, il est évident que les coûts sont non négatifs. De plus, il est possible d'ajouter, au coût de chaque arc, une valeur strictement positive très faible (par rapport aux coûts des arcs) afin de n'avoir que des coûts strictement positifs. Sous cette hypothèse, les contraintes de connexité (4.10) ne sont plus nécessaires. En effet, tout point (x, y) vérifiant les contraintes (4.11)-(4.19) et (4.21) est une solution de (4.23). Résoudre le 1-PRLPA unitaire revient alors à résoudre le programme linéaire en nombres entiers suivant

$$\min\{c^T y \mid (x, y) \in \{0, 1\}^{m+m_K} : (x, y) \text{ satisfait (4.11),(4.12),(4.14)-(4.21)}\}. \quad (5.117)$$

Par ailleurs, on suppose que les coûts des arcs vérifient les inégalités métriques. Ceci permet alors d'utiliser une heuristique que nous introduisons dans le chapitre 6 pour obtenir une borne supérieure. Cette heuristique est appliquée une seule fois au début de l'algorithme.

Nous donnons, dans ce qui suit, un aperçu général de l'algorithme de coupes et branchements puis décrivons les procédures de séparation des contraintes de connexité des demandes et de vulnérabilité que nous utilisons.

5.2.1.1 Aperçu général de l'algorithme

La formulation (5.117) contient deux familles de contraintes en nombre exponentiel : les contraintes de connexité des demandes (4.14) et les contraintes de vulnérabilité (4.21), les autres apparaissant en nombre polynomial. Ces deux familles seront donc ajoutées dynamiquement au programme linéaire courant à l'aide de procédures de séparation. Notre algorithme commence par résoudre le programme linéaire formé par les contraintes (4.11),(4.12),(4.15)-(4.20), *i.e.*,

$$\begin{aligned}&\min\{c^T y \mid (x,y) \in \mathbb{R}^{m+m_K} : (x,y) \text{ satisfait } (4.11),(4.12),\\ &(4.15)-(4.19),(4.20)\}.\end{aligned} \tag{5.118}$$

La solution optimale $(x,y) \in \mathbb{R}^{m+m_K}$ de (5.118) est réalisable pour le problème 1-PRLPA unitaire si (x,y) est un vecteur entier qui satisfait les contraintes de connexité des demandes (4.14) et les contraintes de vulnérabilité (4.21). En général, la solution ainsi obtenue n'est pas réalisable pour le problème 1-PRLPA unitaire. Dans ce cas, à chaque itération, l'algorithme génère des inégalités valides pour le problème et violées par la solution optimale courante. Les inégalités ainsi trouvées sont ajoutées au programme linéaire courant et le nouveau programme est résolu. Ce processus est repris jusqu'à ce qu'aucune inégalité violée ne soit détectée. Si la solution n'est encore pas réalisable, l'algorithme procède alors à un branchement. L'algorithme 7 ci-dessous illustre les principales phases de notre algorithme de coupes et branchements.

On peut remarquer que toutes les inégalités sont globales (*i.e.* valides dans tout l'arbre de branchement) puisque ces dernières sont des contraintes de la formulation. Par ailleurs, plusieurs contraintes peuvent être ajoutées à chaque itération. De plus, on recherche des contraintes de vulnérabilité violées uniquement si aucune contrainte de connexité des demandes n'a été trouvée. Comme ces contraintes sont nécessaires pour déterminer si une solution est réalisable, la séparation de ces contraintes est appliquée à chaque nœud de l'arbre de branchement.

Lorsque des contraintes (4.14) et (4.21) du programme linéaire ne sont plus serrées par la solution (x,y) pendant plusieurs itérations, ces dernières sont supprimées du programme linéaire afin de ne pas alourdir inutilement la taille de ce dernier. Cependant, il est possible que ces contraintes soient par la suite de nouveau violées par la solution courante. Afin de ne pas générer plusieurs fois la même contrainte, chaque contrainte violée trouvée est ajoutée dans un pool. La séparation de chaque famille d'inégalités consiste alors à vérifier d'abord s'il existe des contraintes dans le pool qui soient violées par la solution courante. Si de telles contraintes existent, alors elles sont ajoutées au programme linéaire courant et la séparation s'arrête. Dans le cas contraire, la procédure

Algorithme 7: Algorithme de coupes et branchements

Entrée : un graphe orienté $D = (V, A)$, un sommet v_0 de V, un ensemble de demandes K, un vecteur coût $c \in \mathbb{R}^A$

Sortie : Solution optimale de (5.117)

Début

1 : PL $\leftarrow$ (5.118)

2 : Résoudre le programme linéaire PL.

Soit (x, y) la solution optimale de PL.

3 : **si** (x, y) *est réalisable pour* $P_U(D, v_0, K)$ **alors**

  (x, y) est une solution optimale. STOP

4 : **si** *des contraintes violées par* (x, y) *sont trouvées* **alors**

  Les ajouter à PL.

  Aller en 2.

5 : **sinon**

  Brancher sur une variable fractionnaire.

6 : Prendre la meilleure solution de tous les sous-problèmes.

Fin

de séparation définie ci-après est exécutée pour trouver de nouvelles inégalités violées qui n'aient encore jamais été générées par l'algorithme.

Dans ce qui suit, nous décrivons les procédures de séparation utilisées dans notre algorithme. La solution à séparer sera notée $(\bar{x}, \bar{y})$.

5.2.1.2 Séparation des contraintes de connexité des demandes

Comme indiqué dans le chapitre 4, la séparation des contraintes de connexité des demandes (4.14) se ramène à résoudre np problèmes de coupe minimum. Cependant, pour séparer ces contraintes de manière plus rapide, nous développons une heuristique. Celle-ci permet de séparer ces contraintes d'une manière exacte si le vecteur $\bar{x}$ est entier. Cette heuristique, décrite par l'algorithme 8, peut être présentée de la manière suivante.

L'heuristique de séparation restreint la recherche de contraintes (4.14) violées à celles où l'ensemble de sommets W et le sommet v appartiennent à deux composantes connexes différentes du graphe D^k, correspondant au graphe induit par l'ensemble d'arcs $\{a \in A^k : \bar{x}^k_a > 0\}$. Pour cela, étant donnée une demande $k \in K$, l'algorithme effectue, à partir du sommet o^k, un parcours en profondeur dans le graphe D^k. Soit

Algorithme 8: Heuristique de séparation des contraintes (4.14)

Entrée : vecteur $\bar{x} \in \mathbb{R}^{m_K}$
Sortie : Ensemble S de contraintes de connexité des demandes violées par $\bar{x}$
Début
  $S = \emptyset$;
  pour chaque *demande* $k \in K$ **faire**
    Soit D^k le graphe induit par $\{a \in A^k : \bar{x}^k_a > 0\}$;
    Parcours en profondeur à partir de o^k dans D^k;
    Soit W l'ensemble des sommets visités lors de ce parcours;
    pour chaque *sommet* $v \in V \setminus W$ **faire**
      si $\bar{x}^k(\delta(v) \cap A^k) > 0$ **alors**
        Ajouter dans S la contrainte (4.14) associée à W et v;
  retourner *S;*
Fin

W l'ensemble des sommets pouvant être atteints à partir de o^k dans D^k. On a alors $\bar{x}(\delta^{\text{out}}(W) \cap A^k) = 0$. Considérons un sommet v de $V \setminus W$. Si $\bar{x}^k(\delta(v) \cap A^k) > 0$, cela implique que v appartient à D^k. De plus, ce dernier contient au minimum deux composantes connexes, l'une contenant les sommets de W et l'autre contenant le sommet v. D^k n'est donc pas connexe et la contrainte de connexité des demandes associée à k, W et v est violée par la solution $\bar{x}$. Si pour tous les sommets $v \in V \setminus W$, $\bar{x}^k(\delta(v) \cap A^k) = 0$ alors le graphe D^k est connexe. Comme l'heuristique limite la recherche de contraintes violées à celles dont l'ensemble W et le sommet v appartiennent à deux composantes connexes distinctes, aucune contrainte de connexité des demandes associée à k n'est retournée par l'heuristique.

Cette heuristique de séparation est beaucoup plus rapide que l'algorithme de séparation exact donné dans le chapitre 4 puisqu'elle correspond à p parcours de graphe (largeur ou profondeur). Comme la complexité d'un parcours en profondeur est en $O(m)$, l'heuristique est en $O(mp)$.

Un autre intérêt pour cette procédure de séparation est que si aucune contrainte de connexité des demandes n'a été trouvée, alors les graphes D^k associés aux demandes $k \in K$ sont connexes. Par le lemme 4.23, la séparation des contraintes de vulnérabilité est alors exacte.

5.2.1.3 Séparation des contraintes de vulnérabilité

La procédure de séparation des contraintes de vulnérabilité (4.21) est celle donnée dans le lemme 4.23 du chapitre 4. Comme cette séparation est lancée si aucune contrainte de connexité des demandes n'a été trouvée, alors les graphes D^k associées aux demandes $k \in K$ sont connexes. Comme $\bar{x}$ est non négatif du fait des contraintes triviales (4.18), le lemme 4.23 assure que la séparation des contraintes de vulnérabilité (4.21) est exacte. Cette séparation est décrite par l'algorithme 9. L'algorithme contracte l'origine o^k et la destination d^k en un sommet v^k pour chaque demande $k \in K$. Ceci permet de limiter la recherche aux sous-ensembles W tels que $\delta_\Phi(W) = \emptyset$. Il construit ensuite le graphe auxiliaire $\tilde{D}$ dans lequel un arc (v^k, v) est ajouté pour toute demande $k \in K$ et pour tout sommet v tel que $\bar{x}^k(\delta(v) \cap A^k) > 0$. Un coût infini est associé à chacun de ces arcs. Le lemme 4.23 montre alors que le problème de séparation se ramène à déterminer une v_0v^k-coupe minimum dans $\tilde{D}$ pour toute demande $k \in K$.

Pour calculer une coupe minimum, on utilise l'algorithme de calcul de flot maximum d'Edmonds-Karp [34] dont la complexité est en $O(m^2n)$. L'implémentation de cet algorithme de calcul de flot maximum se fait à travers l'utilisation de la bibliothèque Boost Graph Library (BGL) [92] qui implémente déjà cet algorithme.

Dans le chapitre 4, on a montré que les contraintes de vulnérabilité sont dominées par les contraintes de vulnérabilité renforcées (4.24). Celles-ci correspondent aux contraintes de vulnérabilité dans lesquelles les variables associées aux demandes de $A_\Phi(W)$ sont supprimées. Par conséquent, dès qu'une contrainte de vulnérabilité (4.21), associée à un sous-ensemble W et violée par $(\bar{x}, \bar{y})$, est trouvée par notre procédure de séparation, la contrainte de vulnérabilité renforcée (4.24) associée au sous-ensemble W est ajoutée dans le programme linéaire courant, à la place de la contrainte de vulnérabilité trouvée.

5.2.2 Résultats expérimentaux

5.2.2.1 Contexte informatique

Avant de présenter nos différents résultats expérimentaux, nous donnons un bref descriptif des logiciels et du matériel informatique que nous avons utilisés.

L'algorithme de coupes et branchements décrit précédemment a été implementé en utilisant la librairie CBC (COIN-OR Branch and Cut), version 2.3.1. Pour la résolution des programmes linéaires, CBC fait appel à la librairie CLP (COIN-OR Linear Programing), version 1.10.1, qui est une implémentation rapide de l'algorithme du simplexe.

Algorithme 9: Séparation des contraintes (4.21)

Entrée : vecteur $(\bar{x}, \bar{y}) \in \mathbb{R}^{m_K} \times \mathbb{R}^m$
Sortie : Ensemble S de contraintes de vulnérabilité violées par $(\bar{x}, \bar{y})$
Début
  $S = \emptyset$;
  Soit $\hat{D} = (\hat{V}, \hat{A})$ obtenu par contraction de o^k et d^k en v^k, pour tout $k \in K$;
  pour chaque *demande* $k \in K$ **faire**
    $B^k = \emptyset$;
    pour chaque *sommet* $v \in V \setminus \{v^k\}$ **faire**
      si $\bar{x}^k(\delta(v) \cap A^k) > 0$ **alors**
        $B^k = B^k \cup \{(v^k, v)\}$;
  $\tilde{D} = (\hat{V}, \tilde{A})$, avec $\tilde{A} = \left(\cup_{k \in K} B^k\right) \cup \hat{A}$;
  Détermination des poids $\tilde{w}_a, a \in \tilde{A}$;
  pour chaque *demande* $k \in K$ **faire**
    Soit W tel que $\delta^{\text{out}}(W)$ est la $v_0 v^k$-coupe minimum dans $\tilde{D}$ avec les poids $\tilde{w}$;
    si $\tilde{w}(\delta^{\text{out}}(W)) < 1$ **alors**
      Remplacer les sommets v^k par o^k et d^k dans W;
      Ajouter dans S la contrainte de vulnérabilité renforcée 4 associée à W;
  retourner *S;*
Fin

Ces deux librairies appartiennent au projet COIN-OR (COmputational INfrastructure for Operations Research) [61] qui regroupe un ensemble de librairies libres liées à la recherche opérationnelle.

Ce travail expérimental, implémenté en langage C++, a été réalisé sur un processeur cadencé à 2,5 Ghz avec 6 Go de mémoire vive. Nous avons fixé un temps maximum d'exécution à 2 heures. Dans la section suivante, nous allons présenter brièvement les instances sur lesquelles nous avons effectué nos tests.

5.2.2.2 Description des instances testées

Les résultats qui sont présentés ici ont été obtenus à partir d'instances issues de la TSP Library [82]. Pour toutes les instances, le graphe D, représentant le réseau, est considéré complet. L'ensemble K est choisi de manière à ce que chaque sommet soit

incident à au plus une demande, le dépôt v_0 étant incident à aucune demande. Par ailleurs, nous avons testé des instances avec des coûts symétriques et d'autres avec des coûts asymétriques.

Les instances proviennent de la TSP Library [82]. Les graphes D sont obtenus en considérant des sous-ensembles de sommets des instances. L'ensemble de demandes est généré aléatoirement. Pour chaque taille d'instance, définie par le nombre de sommets et le nombre de demandes, nous avons testé cinq instances. Le graphe D et le vecteur c sont les mêmes pour chacune de ces instances, seules les origines et destinations des demandes changent. Nous considérons les moyennes des résultats obtenus pour ces 5 instances.

Nos résultats sont reportés dans les tables des sections suivantes. Les différentes colonnes de ces tables représentent :

- n : nombre de sommets de l'instance
- p : nombre de demandes de l'instance
- opt : Nombre d'instances résolues à l'optimum sur le nombre d'instances testées
- N_{CD} : Nombre de contraintes de connexité demandes violées générées
- N_{Vuln} : Nombre de contraintes de vulnérabilité violées générées
- N_T : Nombre de sommets dans l'arbre de branchement
- Gap : Erreur relative entre le coût de la meilleure solution obtenue (si l'instance a été résolue à l'optimum, alors cette solution correspond à la solution optimale) et la valeur obtenue avant branchement
- T_{CD} : Temps de séparation des contraintes de connexité en secondes
- T_{Vuln} : Temps de séparation des contraintes de vulnérabilité en secondes
- TT : Temps de résolution de l'algorithme en secondes

5.2.2.3 Instances symétriques

Notre première série d'expérimentations concerne les instances dont les coûts sont symétriques. Nous considérons des instances données par des graphes ayant jusqu'à 101 sommets et 50 demandes. La table 5.1 rapporte la moyenne des résultats obtenus pour ces instances. On remarque que pour les instances contenant moins de 50 demandes, toutes les instances sauf trois ont été résolues à l'optimum durant le temps imparti. Par contre, aucune instance avec 50 demandes n'a été résolue. Pour ces instances, même la relaxation linéaire (5.118) n'a pas pu être résolue dans le temps imparti.

On peut également remarquer que, pour la plupart des instances, un nombre significatif de contraintes de vulnérabilité a été généré alors que peu de contraintes de connexité des demandes ont été trouvées. La séparation des contraintes de connexité des demandes est très rapide (de l'ordre de quelques secondes) et celle des contraintes de vulnérabilité est relativement rapide (au maximum une minute et demi). L'erreur relative entre le coût de la meilleure solution et la valeur de solution avant branchement (Gap) est très faible ; elle est de moins de 0.3%. Cela signifie que la formulation est très serrée et l'on peut espérer que le nombre de nœuds dans l'abre de branchements soit relativement réduit. Ceci est confirmé par les résulats expérimentaux. En effet, près d'une instance sur cinq est résolue de manière optimale sans branchements. De plus, toutes les instances, sauf celles contenant 91 sommets et entre 5 et 15 demandes, sont résolues avec, en moyenne, moins de 15 sommets dans l'arbre de branchements.

5.2.2.4 Instances asymétriques

La table 5.2 donne les résultats pour les instances dont les coûts ne sont pas symétriques. On peut noter que toutes les instances sauf six ont été résolues à l'optimum. En effet, deux instances à 81 sommets et 15 et 25 demandes respectivement, n'ont pas pu être résolues en deux heures de calcul. De même, quatre instances avec 101 sommets et respectivement 10, 30, 35 et 35 demandes n'ont pas pu être résolues à l'optimum. Le nombre moyen de contraintes de connexité des demandes générées est relativement faible. Hormis pour les instances à 81 sommets et 5 demandes, pour lesquelles le nombre moyen de contraintes trouvées est 123, au plus 30 contraintes de connexité des demandes sont trouvées en moyenne. Le temps de séparation de ces contraintes est aussi très faible. Un nombre significatif de contraintes de vulnérabilité sont générées par l'algorithme. Le temps de séparation reste aussi relativement faible au vu du nombre de contraintes générées. On remarque que l'erreur relative entre la meilleure solution trouvée et la valeur de la solution avant branchement ne dépasse pas 1% dans les instances traitées. De plus, le nombre de nœuds dans l'arbre de branchement est réduit.

Ces résultats expérimentaux montrent que la relaxation linéaire est très serrée. Cela s'explique par le nombre de nœuds réduit dans l'arbre de branchements. De plus, l'erreur relative entre la valeur avant branchement et la valeur optimale est très faible. Finalement, beaucoup d'instances sont résolues sans branchement.
Malgré cela, les temps de résolution restent relativement longs. On peut remarquer, par exemple, qu'il a fallu en moyenne plus de 4000 secondes pour résoudre les instances asymétriques de 101 sommets et 50 demandes alors que l'arbre de branchement ne contient que très peu de sommets, et le nombre de contraintes de vulnérabilité violées

générées est très réduit. En fait, la plupart du temps de calcul est utilisé dans la résolution de la relaxation linéaire de (5.118). Bien que le nombre de contraintes et de variables soit polynomial, il devient difficile de résoudre efficacement cette relaxation linéaire lorsque la taille des instances augmente.

5.3 Polyèdre du 1-PRLPA unitaire dans les cactus

Dans cette section, nous caractérisons d'abord le polytope des solutions du 1-PRLPA unitaire quand le graphe est réduit à un circuit et chaque sommet différent du dépôt est incident à au moins une demande. Nous étendons ensuite cette description polyédrale aux cactus lorsque les demandes vérifient certaines hypothèses. Soit (C_n, v_0, K) une instance du 1-PRLPA unitaire où C_n est un circuit à n sommets doublement orienté, *i.e.*, il existe deux arcs (u, v) et (v, u) entre deux sommets u et v consécutifs.

5.3.1 Notations

Avant de commencer cette description polyédrale, on introduit quelques définitions et notations. Si $C_n = (V, A)$ est un circuit doublement orienté à n sommets, alors on note les sommets de V par $\{v_0, v_1, \ldots, v_{n-1}\}$. Pour une question de clarté, dans le reste de cette section, le sommet v_0 sera parfois noté v_n. Les sommets v_i et v_{i+1} sont adjacents pour tout $i = 1, 2, \ldots, n-1$. Sans perte de généralité, on suppose que la numérotation des sommets du circuit satisfait les conditions suivantes :

C1 : s'il existe des demandes non incidentes à v_0, alors il en existe au moins une, disons k, telle que $o^k = v_i$ et $d^k = v_j$, avec $0 < i < j < n$,

C2 : si toutes les demandes sont incidentes à v_0 et si v_0 est à la fois origine et destination d'une demande, alors il existe deux demandes k et k', avec $v_0 = o^k$ et $v_0 = d^{k'}$, telles que $d^k = v_i$, $o^{k'} = v_j$ et $0 < i < j < n$.

L'ensemble d'arcs A est tel que $A = A^+ \cup A^-$ avec

$$A^+ = \{(v_i, v_{i+1}) : i = 0, 1, \ldots, n-1\}$$

et

$$A^- = \{(v_{i+1}, v_i) : i = 0, 1, \ldots, n-1\}.$$

On remarque que, pour tout couple de sommets distincts u et v de V, il n'existe que deux chemins élémentaires distincts allant de u à v. Cela implique que pour chaque demande, il n'existe que deux chemins possibles pour transporter la demande de son

origine à sa destination. Considérons une demande $k \in K$. Si les deux chemins élémentaires arc-disjoints sont notés par L^1 et L^2, il est clair que toute solution réalisable du 1-PRLPA unitaire est telle que $x_a^k = 0$ pour tout arc $a \in A^k \setminus (L^1 \cup L^2)$. On redéfinit alors l'ensemble d'arcs A^k de chaque demande correspondant à l'ensemble d'arcs des deux chemins de o^k à d^k. L'entier m_k correspond toujours à la cardinalité de A^k pour toute demande $k \in K$ et est donc égal au nombre d'arcs des deux chemins élémentaires de o^k à d^k.

Les contraintes de circuit (4.13) et les contraintes de connexité des demandes (4.14) ne sont plus utiles dans la formulation. En effet, les premières correspondent alors aux contraintes triviales (4.19) puisque pour toute demande $k \in K$, tout sommet de $V \setminus \{o^k, d^k\}$ est incident à exactement un arc de A^k. Les contraintes (4.14) sont toujours vérifiées puisque le point x satisfait les contraintes de conservation de flot (4.12) et que l'ensemble A^k forme un graphe acyclique pour toute demande $k \in K$. De plus, on considère ici les contraintes de vulnérabilité renforcées (4.24) à la place des contraintes de vulnérabilité (4.21) puisque ces contraintes sont réalisables pour le 1-PRLPA unitaire et dominent les contraintes de vulnérabilité. L'ensemble des solutions réalisables pour le 1-PRLPA unitaire quand le graphe est un circuit doublement orienté correspond alors à l'ensemble

$$\{(x, y) \in \{0,1\}^{m_K} \times \{0,1\}^m : (x, y) \text{ satisfait } (4.10) - (4.12), (4.15) - (4.19), (4.24)\}$$

Le polytope associé à l'instance (C_n, v_0, K) est noté $P_U(C_n, v_0, K)$. La démonstration donnant la description de $P_U(C_n, v_0, K)$ dépend de la position des demandes entre elles et par rapport au sommet v_0. Pour cela, on utilise la notion de circuit réduit. Étant donnée une instance (C_n, v_0, K), on dit que C' est un *circuit réduit* de (C_n, v_0, K) si C' peut être obtenu à partir de (C_n, v_0, K) à l'aide des opérations suivantes :

- O_1 : identifier les extrémités d'un arc et supprimer les boucles,
- O_2 : supprimer une demande.

5.3.2 Inégalités valides

Nous allons maintenant décrire de nouvelles contraintes valides pour $P_U(C_n, v_0, K)$. Soit une partition $\pi = \{W_1, W_2, \ldots, W_s\}$ de V telle que $v_0 \in W_1$, $\{o^k, d^k : k \in K\} \cap W_i \neq \emptyset$ pour tout $i = 2, \ldots, s$ et W_i est composé d'un chemin pour tout $i = 1, 2, \ldots, s$. Autrement dit, le graphe obtenu par contraction des sous-ensembles W_i de π correspond à un circuit. Soit

$$A_\pi = ([W_2 : W_3 : \ldots : W_s] \cap A^+) \cup \delta^{\text{out}}(W_1).$$

Notons par a_0 l'arc de $\delta^{\text{out}}(W_1) \cap A^-$. On note par K_π l'ensemble des demandes $k \in K$ ayant leur deux extrémités dans un même sous-ensemble W_i, $i = 1, 2, \ldots, s$ et telles que l'un des deux chemins élémentaires de o^k à d^k passe par a_0. La contrainte

$$y(A_\pi) - x_{a_0}(K_\pi) \geq s - 1 \tag{5.119}$$

associée à la partition π est appelée *contrainte de partition*.

Considérons aussi la contrainte

$$y_{(v_0,v_1)} - y_{(v_1,v_0)} \geq 0. \tag{5.120}$$

Cette contrainte sera appelée *contrainte de priorité*. En effet, elle exprime le fait que l'arc (v_1, v_0) ne peut être traversé par le véhicule que si ce dernier traverse également l'arc (v_0, v_1).

Proposition 5.14 *Les contraintes de partition* (5.119) *sont valides pour* $P_U(C_n, v_0, K)$. *De plus, si* K *n'est pas contenu dans l'ensemble* $\{(v_0, v), (v, v_0)\}$, *avec* v *un sommet de* $V \setminus \{v_0\}$, *alors la contrainte de priorité* (5.120) *est aussi valide pour* $P_U(C_n, v_0, K)$.

Preuve. Considérons d'abord la contrainte de priorité (5.120) et supposons que K n'est pas contenu dans l'ensemble $\{(v_0, v), (v, v_0)\}$, avec v un sommet de $V \setminus \{v_0\}$. Supposons que cette contrainte n'est pas valide, *i.e.*, il existe un point (x, y) de $P_U(C_n, v_0, K)$ tel que $y_{(v_0,v_1)} - y_{(v_1,v_0)} = -1$. Par les contraintes de conservation de flot, on a alors $y_a = 1$ pour tout arc $a \in A^-$ et $y_a = 0$ pour tout $a \in A^+$. Supposons qu'il existe au moins une demande qui n'est pas incidente à v_0. D'après la condition C1 de la numérotation des sommets de V, il existe alors $k \in K$ telle que $o^k = v_i$, $d^k = v_j$ et $0 < i < j < n$. Les contraintes de capacité impliquent que $x^k_{(v_0,v_{n-1})} = 1$. La contrainte de vulnérabilité renforcée (4.24) associée à v_0 est alors violée par (x, y), une contradiction.

Supposons maintenant que toutes les demandes de K sont incidentes au dépôt. Si v_0 est l'origine ou la destination d'au moins deux demandes, alors tout point réalisable de $P_U(C_n, v_0, K)$ doit satisfaire $y_{(v_0,v_1)} = y_{(v_0,v_{n-1})} = 1$. Les contraintes de conservation de flot associées au sommet v_0 impliquent que la contrainte de priorité est satisfaite. Supposons finalement que v_0 est l'origine et la destination d'au plus une demande. Alors $K = \{1, 2\}$. D'après la condition sur K, on a $v_0 = o^1 = d^2$, $d^1 = v_i$ et $o^2 = v_j$, avec $1 < i < j < n$. Les chemins des demandes 1 et 2 intersectant A^- passent tous les deux par l'arc (v_0, v_{n-1}), ce qui implique que (x, y) n'est pas valide. L'inégalité de priorité est donc valide pour $P_U(C_n, v_0, K)$.

On montre maintenant la validité des contraintes de partition. Considérons une inégalité définie par une partition $\pi = \{W_1, W_2, \ldots, W_s\}$. Considérons un point (x, y) de $P_U(C_n, v_0, K)$. Supposons d'abord que $y_{(i,j)} = y_{(j,i)}$ pour tout arc (i, j) de A. Considérons le graphe non orienté $\hat{G} = (\hat{V}, \hat{E})$ où $\hat{V}$ est l'ensemble des sommets obtenus par contraction des sous-ensembles W_i, $i = 1, 2, \ldots, s$ et $\hat{E}$ est l'ensemble des arêtes $\{i, j\}$ tels que $y([W_i : W_j]) \geq 1$. $\hat{E}$ est l'ensemble d'arêtes obtenu suite aux contractions des W_i. Supposons d'abord que $x_{a_0}(K_\pi) = 0$. Comme (x, y) est réalisable, le graphe $\hat{G}$ est connexe. On a donc $|\hat{E}| \geq s - 1$. Par ailleurs, pour toute arête $\{i, j\}$ de $\hat{E}$, il existe un arc a avec $y_a = 1$ appartenant soit à $[W_2 : W_3 : \ldots : W_s] \cap A^+$ soit à $\delta^{\text{out}}(W_1)$. La contrainte de partition est donc vérifiée. Supposons maintenant que $x_{a_0}(K_\pi) = 1$. Ceci implique qu'il existe une demande $k \in K_\pi$ passant par tous les arcs de $[W_1 : W_2 : \ldots : W_s] \cap A^-$. Par les contraintes de capacité, on obtient $y_a = 1$ pour tout $a \in [W_1 : W_2 : \ldots : W_s] \cap A^-$. Comme $y_{(i,j)} = y_{(j,i)}$, il en résulte que $y(A_\pi) \geq s$, ce qui implique que la contrainte est vérifiée. Supposons finalement que (x, y) ne vérifie pas les équations $y_{(i,j)} = y_{(j,i)}$ pour tout (i, j) de A. D'après la contrainte de priorité (5.120), il s'ensuit que l'on a $y_a = 1$ pour tout $a \in A^+$ et $y_a = 0$ pour tout $a \in A^-$. Les contraintes de capacité impliquent que l'on a $x_{a_0}(K_\pi) = 0$. Comme A_π contient $s - 1$ arcs appartenant à A^+, la contrainte est donc vérifiée. □

Les contraintes de partition peuvent être étendues aux partitions où les W_i ne sont pas formés par des chemins. Dans ce cas, l'arc a_0 est un des arcs de $\delta^{\text{out}}(W_1) \cap A^-$. Cependant, ces contraintes plus générales sont redondantes par rapport aux contraintes de partition dont les sous-ensembles W_i sont formés par des chemins. En effet, considérons une partition $\pi = \{W_1, W_2, \ldots, W_s\}$ de V telle que $v_0 \in W_1$, $\{o^k, d^k : k \in K\} \cap W_i \neq \emptyset$ pour tout $i = 2, \ldots, s$ et un arc a_0 de $\delta^{\text{out}}(W_1) \cap A^-$. Supposons que les ensembles W_i ne sont pas tous formés par des chemins. Notons par A_π^{-1} l'ensemble d'arcs opposés de A_π, *i.e.*, (i, j) appartient à A_π^{-1} si (j, i) appartient à A_π. Notons par $\pi^1 = \{W_1^1, W_2^1, \ldots, W_{s^1}^1\}$ la partition des sommets de V correspondant aux composantes connexes de $C_n \setminus \{A_\pi \cup A_\pi^{-1}\}$. Supposons qu'il existe $i \in \{2, 3, \ldots, s^1\}$ tel que $W_i^1 \cap \{o^k, d^k : k \in K\} = \emptyset$. Parmi les deux sous-ensembles de π^1 adjacents à W^1, il en existe un, disons W_j^1, tel que $[W_i^1 : W_j^1]$ ne contient pas a_0. Considérons alors la partition π^2 obtenue à partir de π supprimant W_i^1 du sous-ensemble de sommets dans lequel il se trouve pour l'intégrer dans celui contenant W_j^1. On a alors $A_{\pi^2} = A_\pi \setminus \{a\}$, avec $a \neq a_0$. La contrainte associée à π^2 domine donc celle associée à π^1. On peut donc supposer, sans perte de généralité, que W_1^1 contient v_0 et que les ensembles W_i^1, $i = 2, 3, \ldots, s^1$ contiennent au moins une origine ou une destination.
π^1 est composé de sous-ensembles correspondant à des chemins. Il est clair que chaque sous-ensemble W_i, $i = 1, 2, \ldots, s$, est l'union de plusieurs sous-ensembles W_i^1, $i = 1, 2, \ldots, s^1$. Supposons que W_1 est l'union de r sous-ensembles W_i^1 de π^1. L'ensemble

A_{π^1} diffère alors de A_π d'exactement r arcs, notés $(u_1, v_1), (u_2, v_2), \ldots, (u_r, v_r)$ tels que

$$A_{\pi^1} = A_\pi \cup \{(u_1, v_1), (u_2, v_2), \ldots, (u_r, v_r)\} \setminus \{(v_1, u_1), (v_2, u_2), \ldots, (v_r, u_r)\}.$$

Par ailleurs, pour tout arc $(i, j) \in A^+$, on a $y_{(j,i)} - y_{(i,j)} - x_{a_0}(K_\pi) \geq -1$ puisque toute demande de K_π passant par a_0 passe aussi par (j, i). Comme $s^1 \geq s + r$, la contrainte de partition associée à π est dominée par la somme de ces contraintes ainsi que la contrainte de partition associée à π^1. Si r vaut 1, il faut alors additionner la contrainte de capacité $x_{a_0}(K_\pi) \leq 1$ avec la contrainte de partition associée à π^1 pour obtenir celle associée à π. Dans tous les cas, la contrainte de partition associée à π est dominée par les contraintes de partition dont les sous-ensembles sont formés par des chemins. On considère alors par la suite uniquement ces contraintes.

5.3.3 Polyèdre du 1-PRLPA unitaire dans les circuits

Dans cette section, nous caractérisons le polytope $P_U(C_n, v_0, K)$ du 1-PRLPA unitaire lorsque C_n est un circuit doublement orienté et chaque sommet de $V \setminus \{v_0\}$ est incident à au moins une demande. On suppose, dans la suite, que C_n et K satisfont ces deux hypothèses. Le polytope dans le cas $p = 0$ ou dans le cas $n = 2$ étant trivial, on suppose, sans perte de généralité, que $n \geq 3$ et $p \geq 1$. Dans ce cas, la contrainte de priorité (5.120) est valide pour $P_U(C_n, v_0, K)$. On introduit maintenant une proposition qui sera utile par la suite pour prouver qu'une solution (x, y) appartient à $P_U(C_n, v_0, K)$.

Proposition 5.15 *Soit $(x, y) \in \mathbb{R}^{m_K} \times \mathbb{R}^m$ un point en 0-1 vérifiant les contraintes* (4.10)-(4.12) *et* (4.15)-(4.19)*. Si (x, y) vérifie la contrainte de vulnérabilité renforcée* (4.24) *associée à v_0, alors (x, y) appartient à $P_U(C_n, v_0, K)$.*

Preuve. Il suffit de montrer que si (x, y) vérifie la contrainte de vulnérabilité renforcée (4.24) associée à v_0, alors il vérifie toutes les contraintes (4.24). Supposons qu'il existe une contrainte de vulnérabilité renforcée associée à W différente de v_0 qui soit violée par (x, y). On montre alors que la contrainte de vulnérabilité associée à v_0 est également violée par (x, y). Supposons, sans perte de généralité, que W est composé de plusieurs sous-ensembles non adjacents, disons $W_1, W_2, \ldots, W_t$. Comme (x, y) vérifie les contraintes de capacité, il s'ensuit que pour tout arc $a \in \delta^{\text{out}}(W)$ avec $y_a = 1$, on a $\sum_{k \in A_\phi(\overline{W}) \cap K^a} x_a^k = 1$. Par ailleurs, pour tout $i = 1, 2, \ldots, t$, on a $\delta^{\text{out}}(W_i) \subseteq \delta^{\text{out}}(W)$ et $A_\phi(\overline{W}) \subseteq A_\phi(\overline{W}_i)$, ce qui implique que la contrainte associée à W_i est violée par (x, y). Supposons que W_1 contient v_0. Notons par (u_1, v_1) et (u_2, v_2) les arcs de $\delta^{\text{out}}(W_1)$. Il est clair que l'on a $y_{(u_1,v_1)} + y_{(u_2,v_2)} \geq 1$ puisque (x, y) vérifie les contraintes de

connexité. Ceci implique qu'il existe une demande ayant ses deux extrémités dans $\overline{W}_1$ passant par (u_1, v_1) ou (u_2, v_2). Comme les chemins des demandes sont élémentaires, on a alors $y_{(u_1,v_1)} + y_{(v_1,u_1)} \geq 1$ et $y_{(u_1,v_1)} + y_{(v_2,u_2)} \geq 1$. Les contraintes de conservation de flot assurent alors que $y(\delta^{\text{out}}(W_1)) = y(\delta^{\text{out}}(v_0))$, ce qui implique que la contrainte de vulnérabilité renforcée (4.24) associée à v_0 est violée par (x, y) □

Cette proposition permet de déduire que si le dépôt v_0 est incident à une demande, disons k, alors toutes les contraintes de vulnérabilité renforcées (4.24) sont satisfaites pour tout point (x, y) satisfaisant les autres contraintes de la formulation. En effet, il est facile de voir que la contrainte de vulnérabilité renforcée associée à v_0 est violée si tous les arcs incidents à v_0 avec $y_a = 1$ appartiennent au chemin d'une demande ayant ses deux extrémités dans $V \setminus \{v_0\}$. Par ailleurs, comme v_0 est l'origine ou la destination de k, il existe un arc a incident à v_0 appartenant au chemin de la demande k. Comme les chemins des demandes sont arc-disjoints, a n'appartient au chemin d'aucune demande ayant ses deux extrémité dans $V \setminus \{v_0\}$. La contrainte de vulnérabilité renforcée (4.24) associée à v_0 est satisfaite par (x, y). La proposition 5.15 implique alors que toutes les contraintes de vulnérabilité renforcées sont satisfaites par (x, y).

On donne maintenant la description du polytope $P_U(C_n, v_0, K)$. Pour cela, on introduit un premier circuit réduit à deux sommets, appelé *circuit double origine/destination*, représenté dans la figure 5.1. Ce circuit comporte deux demandes distinctes $k \neq k'$, ayant la même origine et la même destination. On dit alors que les demandes k et k' forment un circuit double origine/destination.

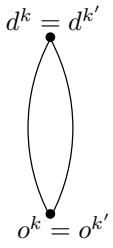

FIGURE 5.1 – Circuit double origine/destination

Le théorème suivant décrit le polytope des solutions du 1-PRLPA unitaire dans le cas où le graphe est un circuit doublement orienté dont tous les sommets différents du dépôt sont incidents à au moins une demande.

Théorème 5.16 *$P_U(C_n, v_0, K)$ est donné par les contraintes* (4.10)-(4.12),(4.15)-(4.19)*,* (4.24),(5.119) *et* (5.120)*.*

Preuve. La preuve est divisée en plusieurs parties, elle dépend du positionnement des demandes sur le circuit. On s'intéresse pour le moment au cas où C_n peut être réduit (par les opérations O_1 et O_2) à un circuit double origine/destination par rapport à deux demandes, disons 1 et 2. On suppose que les origines et destinations des demandes 1 et 2 sont toutes distinctes. De plus, on suppose que o^2 et d^2 appartiennent au même chemin de o^1 à d^1. Les preuves des autres cas sont similaires et ne seront donc pas données. On note par V_1 (respectivement V_2) l'ensemble des sommets du chemin de o^1 à o^2 (respectivement d^1 à d^2) ne passant pas par d^1 ni d^2 (respectivement o^1 ni o^2). On note également par P_1 (respectivement P_2) le chemin de o^1 à d^1 (respectivement o^2 à d^1) ne passant pas par o^2 (respectivement o^1). On suppose, sans perte de généralité, qu'il n'existe pas deux demandes distinctes $k \neq k'$ de $K \setminus \{1,2\}$ avec o^k et $o^{k'}$ appartenant à V_2, d^k et $d^{k'}$ appartenant à V_1 et telles que l'on a $\{o^k, o^{k'}\} \not\subseteq \{d^1, d^2\}$ ou $\{d^k, d^{k'}\} \not\subseteq \{o^1, o^2\}$. Dans le cas contraire, les deux demandes considérées dans la preuve sont les demandes k et k' au lieu des demandes 1 et 2.

Les assertions suivantes décrivent certaines conséquences du fait que les demandes 1 et 2 forment un circuit double origine/destination.

Assertion 5.17 : Les équations $y_{(i,j)} = y_{(j,i)}$ pour tout $(i,j) \in A$ sont satisfaites dans $P_U(C_n, v_0, K)$.

Preuve de l'assertion 5.17 : Comme les demandes 1 et 2 forment un circuit double origine/destination, les chemins de l'origine à la destination des demandes 1 et 2 traversent un arc sortant de V_1. On a alors $x^1(\delta^{\text{out}}(V_1)) = x^2(\delta^{\text{out}}(V_1)) = 1$. Par les contraintes de capacité (4.15), on obtient $y(\delta^{\text{out}}(V_1)) = 2$. Les contraintes de conservation de flot associées à y impliquent que l'on a également $y(\delta^{\text{in}}(V_1)) = 2$. Comme $\deg^{\text{in}}(V_1) = \deg^{\text{out}}(V_1) = 2$, il existe un arc (i,j) de $\delta^{\text{out}}(V_1)$ tel que $y_{(i,j)} = y_{(j,i)} = 1$. D'après les contraintes de conservation associées à y et comme C_n est un circuit doublement orienté, il s'ensuit que les équations $y_{(i,j)} = y_{(j,i)}$ pour tout $(i,j) \in A$ sont satisfaites par tout point (x,y) de $P_U(C_n, v_0, K)$. □

Pour toute demande $k \in K \setminus \{1,2\}$, notons par L_k^1 un $o^k d^k$-chemin élémentaire n'intersectant ni P_1, ni P_2.

Assertion 5.18 : Tout point $Z = (x,y)$ de $P_U(C_n, v_0, K)$ vérifie

$$x_a^k = \begin{cases} 1 & \text{si } a \in L_k^1, \\ 0 & \text{sinon,} \end{cases} \qquad \forall\, k \in K \setminus \{1,2\}, o^k \notin V_2 \text{ ou } d^k \notin V_1 \tag{5.121}$$

$$y_a = 1 \qquad \forall\, a \in P_1 \cup P_2. \tag{5.122}$$

$$x_a^1 = x_{a'}^2 \qquad \forall\, a \in P_i, \forall\, a' \in P_j, i \neq j \in \{1,2\} \tag{5.123}$$

Preuve de l'assertion 5.18 : Par la définition de l'ensemble V_1, on a $\deg^{\text{out}}(V_1) = 2$. Notons par a_1 et a_2 les arcs sortant de V_1. Comme les demandes 1 et 2 ont leur origine dans V_1 et leur destination dans $V \setminus V_1$, il s'ensuit que $x^1(\delta^{\text{out}}(V_1)) + x^2(\delta^{\text{out}}(V_1)) = 2$. D'après les contraintes de capacité (4.15), on déduit que $x^1_a + x^2_a = 1$ pour $a \in \{a_1, a_2\}$. D'après les contraintes de conservation de flot (4.12) associées aux demandes 1 et 2, on obtient $x^1_a + x^2_a = 1$ pour tout $a \in P_1 \cup P_2$. De plus, les contraintes (4.15) impliquent que l'on a également $y_a = 1$ pour tout $a \in P_1 \cup P_2$ et $x^k_a = 0$ pour tout $a \in P_1 \cup P_2$ et pour tout $k \in K^a \setminus \{1, 2\}$. Ceci implique qu'une demande $K \setminus \{1, 2\}$ ne peut être transportée sur un chemin intersectant P_1 ou P_2. Comme $P_U(C_n, v_0, K)$ est non-vide, pour toute demande k de $K \setminus \{1, 2\}$, il existe au moins un $o^k d^k$-chemin élémentaire n'intersectant ni P_1 ni P_2. Par énumération des cas possibles, on remarque que ce chemin est unique pour toute demande de $k \in K \setminus \{1, 2\}$ telle que $o^k \notin V_2$ ou $d^k \notin V_1$, ce qui implique que les équations (5.121) sont vérifiées par (x, y).
On a montré que $y_a = 1$ pour tout $a \in P_1 \cup P_2$. Il s'ensuit, d'après l'assertion 5.17, que (x, y) vérifie également les équations (5.122).
Maintenant, il est facile de voir que chaque $o^k d^k$-chemin élémentaire de la demande k, $k = 1, 2$, contient soit a_1, soit a_2. D'après les contraintes de conservations de flot (4.12) associées à la demande 1, on a alors $x^1_{a_1} = 1 - x^1_{a_2}$. Comme $x^1_{a_2} + x^2_{a_2} = 1$, on obtient $x^1_{a_1} = x^2_{a_2}$. Les contraintes (4.12) impliquent alors que (x, y) vérifie les équations (5.123). □

Si $P_U(C_n, v_0, K)$ est non vide, alors, par les contraintes de capacité, il existe au maximum deux demandes dont l'origine et la destination appartiennent respectivement à V_2 et V_1. On suppose, dans le reste de la preuve du cas d'un circuit restreint double origine/destination, que de telles demandes n'existent pas. La preuve est similaire si de telles demandes existent. Il en résulte que pour tout point $Z = (x, y)$, x^k est donné par les équations (5.121) pour tout $k \in K \setminus \{1, 2\}$.

Si $P_U(C_n, v_0, K)$ est non vide, alors on a

$$x_a(K^a \setminus \{1, 2\}) \leq \begin{cases} 0 & \text{si } a \in P_1 \cup P_2, \\ 1 & \text{sinon,} \end{cases} \qquad \forall\, a \in A. \tag{5.124}$$

Autrement, il n'existerait aucun point (x, y) de $P_U(C_n, v_0, K)$ respectant les contraintes de capacité. De plus, si v_0 n'est incident à aucune demande, alors on a

$$\sum_{k \in K \setminus \{1,2\}} x^k(\delta^{\text{out}}(v_0) \cap A^k) \leq \begin{cases} 0 & \text{si } v_0 \in P_1 \cup P_2, \\ 1 & \text{sinon.} \end{cases} \tag{5.125}$$

En effet, supposons que v_0 est n'est incident à aucune demande. Comme $\deg^{\text{out}}(v_0) = 2$, alors $y(\delta^{\text{out}}(v_0)) \leq 2$. De plus, si v_0 appartient à $P_1 \cup P_2$, alors $x^1(\delta^{\text{out}}(v_0) \cap A^1) +$

$x^2(\delta^{\text{out}}(v_0) \cap A^2) = 1$. L'inégalité (5.125) doit alors être satisfaite pour qu'un point (x, y) de $P_U(C_n, v_0, K)$ satisfasse la contrainte de vulnérabilité renforcée (4.24) associée à v_0.

On distingue maintenant deux cas. Supposons d'abord que $V_1 = \{o^1, o^2\}$, $V_2 = \{d^1, d^2\}$ et $x_a(K \setminus \{1, 2\}) = 0$ pour tout arc a appartenant à $\{(o^1, o^2), (o^2, o^1), (d^1, d^2), (d^2, d^1)\}$. Soit a_1 un arc de P_1. Dans ce cas, si $Z = (x, y)$ est un point de $P_U(C_n, v_0, K)$, alors d'après l'assertion 5.18, Z est identifié si les composantes $y_{(o^1,o^2)}$, $y_{(d^1,d^2)}$ et $x^1_{a_1}$ sont définies. En effet, d'après les équations (5.122), seules les valeurs de y associées aux arcs (o^1, o^2), (o^2, o^1), (d^1, d^2) et (d^2, d^1) ne sont pas définies. D'après l'assertion 5.17, $y_{(o^1,o^2)} = y_{(o^2,o^1)}$ et $y_{(d^1,d^2)} = y_{(d^2,d^1)}$. Si les valeurs $y_{(o^1,o^2)}$ et $y_{(d^1,d^2)}$ sont définies alors y l'est aussi. D'après les équations (5.121), seules les valeurs de x associées aux demandes 1 et 2 ne sont pas définies. Par ailleurs, pour tout $k \in K$, s'il existe un arc $a \in A^k$ tel que x^k_a est défini, alors d'après les contraintes de conservation de flot (4.12), le vecteur x^k est défini. Ceci implique que si $x^1_{a_1}$ est défini, alors x^1 l'est aussi. Finalement, d'après les équations (5.123), il est clair que le vecteur x^2 est déterminé si $x^1_{a_1}$ est connu. Toute solution $Z = (x, y)$ de $P_U(C_n, v_0, K)$ sera donc seulement donnée par les composantes $y_{(o^1,o^2)}$, $y_{(d^1,d^2)}$ et $x^1_{a_1}$.

Soit Z^1 (respectivement Z^2) le point défini par $y_{(o^1,o^2)} = 0$, $y_{(d^1,d^2)} = 1$ (respectivement $y_{(o^1,o^2)} = 1$, $y_{(d^1,d^2)} = 0$) et $x^1_{a_1} = 1$ et soit Z^3 (respectivement Z^4) le point donné par $y_{(o^1,o^2)} = 1$, $y_{(d^1,d^2)} = 1$ et $x^1_{a_1} = 0$ (respectivement $x^1_{a_1} = 1$). Les points Z^i, $i = 1, 2, 3, 4$ appartiennent à $P_U(C_n, v_0, K)$. En effet, considérons le point $Z^1 = (x, y)$. Il est facile de voir que les contraintes de conservation de flot (4.11) et (4.12) et les contraintes triviales (4.16)-(4.19) sont satisfaites. De plus, le graphe induit par l'ensemble d'arcs $\{a \in A : y_a = 1\}$ est connexe, ce qui implique Z^1 satisfait les contraintes de connexité (4.10). D'après les inégalités (5.124), les contraintes de capacité (4.15) sont satisfaites puisque $x^k_a = 0$ pour tout arc $a \in A^k$ n'appartenant pas à P_1 ou P_2, $k = 1, 2$. Considérons maintenant la contrainte de vulnérabilité renforcée (4.24) associée à v_0. Si v_0 est incident à une demande, alors cette contrainte est vérifiée. Supposons maintenant que v_0 n'est incident à aucune demande. Alors v_0 appartient à P_1 ou P_2. On a alors $x^1(\delta^{\text{out}}(v_0) \cap A^1) + x^2(\delta^{\text{out}}(v_0) \cap A^2) = 1$. De plus, d'après (5.122), on a $y(\delta^{\text{out}}(v_0)) = 2$. D'après (5.125), la contrainte de vulnérabilité renforcée associée à v_0 est satisfaite par Z^1. La proposition 5.15 implique alors que Z^1 appartient à $P_U(C_n, v_0, K)$. La preuve pour les autres points Z^i, $i = 2, 3, 4$ est similaire.

Tout point $Z = (x, y)$ peut s'écrire comme combinaison convexe de ces quatre points, *i.e.*,

$$Z = (1 - y_{(o^1,o^2)})Z^1 + (1 - y_{(d^1,d^2)})Z^2 + (1 - x^1_{a_1})Z^3 + (y_{(o^1,o^2)} + y_{(d^1,d^2)} - (1 - x^1_{a_1}) - 1)Z^4.$$

En effet, il est facile de voir que la somme des coefficients est égale à un. On montre maintenant que les coefficients de cette combinaison sont positifs ou nuls. Ceci est clair pour les trois premiers grâce aux contraintes triviales (4.17) et (4.19). Considérons maintenant le coefficient associé au point Z^4. Supposons, sans perte de généralité, que v_0 est un sommet de P_1. Notons par W l'ensemble des sommets de P_1. On a alors $y_{(o^1,o^2)}+y_{(d^1,d^2)} = y(\delta^{\text{out}}(W))$. De plus, la demande 2 a ses deux extrémités dans $A_\Phi(\overline{W})$, et (d^1, d^2) appartient à $A^2 \cap \delta^{\text{out}}(W)$. Finalement, on a $x^2_{(d^1,d^2)} = 1 - x^1_{a^1}$. La contrainte de vulnérabilité renforcée (4.24) associée à W assure alors que le dernier coefficient est positif ou nul.

Nous distinguons maintenant trois cas : V_1 est différent de $\{o^1, o^2\}$, V_2 est différent de $\{d^1, d^2\}$ et le cas où il existe un arc a de $\{(o^1, o^2), (o^2, o^1), (d^1, d^2), (d^2, d^1)\}$ avec $x_a(K^a \setminus \{1, 2\}) = 1$. On montre que, dans ces trois cas, $x^1_{a_1} = 1$. On ne donne la preuve que dans le cas où V_1 est différent de $\{o^1, o^2\}$, la preuve pour les deux autres cas est similaire. Supposons qu'il existe une solution (x, y) avec $x^1_{a_1} = 0$. Tout arc (u, v) de A avec $u, v \in V_1$, est alors traversé par la demande 1 ou la demande 2. Comme $V_1 \neq \{o^1, o^2\}$, il existe un sommet v différent de o^1 et d^1 dans V_1. Le sommet v correspond au dépôt ou est incident à une demande. De plus, on a $x^1(\delta^{\text{out}}(v)) + x^2(\delta^{\text{out}}(v)) = 2$ et $x^1(\delta^{\text{in}}(v))+x^2(\delta^{\text{in}}(v)) = 2$. Comme C_n est un circuit doublement orienté, on a également $\deg^{\text{out}}(v) = 2$ et $\deg^{\text{in}}(v) = 2$. Si $v = v_0$, alors il n'existe pas d'arc a sortant de v_0 avec $y_a = 1$ et $x^a(K^a) = 0$. La contrainte de vulnérabilité renforcée (4.24) associée à v_0 est donc violée par (x, y). Si v est différent du dépôt, alors v est incident à une demande de $K \setminus \{1, 2\}$, disons la demande 3. On a alors $x^3(\delta^{\text{out}}(v)) = 1$ ou $x^3(\delta^{\text{in}}(v)) = 1$. Dans ce cas, les chemins 1,2 et 3 ne sont pas arc-disjoints et il existe alors un arc de $\delta(v)$ pour lequel la contrainte de capacité (4.15) est violée par (x, y). Tout point (x, y) de $P_U(C_n, v_0, K)$ satisfait donc $x^1_{a_1} = 1$. Comme $x^1_{a_1} = x^2_{a_2}$, tout point (x, y) de $P_U(C_n, v_0, K)$ vérifie

$$x^k_a = \begin{cases} 1 & \text{si } a \in P_k, \\ 0 & \text{sinon,} \end{cases} \qquad \forall\ k = 1, 2. \tag{5.126}$$

Notons A_0 le sous-ensemble d'arcs de A^+ tels qu'aucune demande n'est transportée ni sur ces arcs, ni sur leur arcs inverses, c'est-à-dire,

$$A_0 = \{(i, j) \in A^+ : x_{(i,j)}(K^{(i,j)}) = x_{(j,i)}(K^{(j,i)}) = 0\}.$$

Notons A_0^{-1} l'ensemble des arcs inverses de A_0, ie

$$A_0^{-1} = \{(i, j) \in A^- : (j, i) \in A_0\}.$$

Considérons un point $Z = (x, y)$. L'assertion 5.17 et les contraintes de capacité (4.15) impliquent que pour tout arc $a \in A \setminus \{A_0 \cup A_0^{-1}\}$, on a $y_a = 1$. De plus, d'après les

équations (5.121) et (5.126), x est défini. D'après l'assertion 5.17, Z est identifié si les valeurs y_a sont définies pour tout arc $a \in A_0$. Dans ce qui suit, le point Z sera donc seulement donné par les composantes y_a, $a \in A_0$.

Pour tout arc $a' = (i,j) \in A_0$, on note par $Z_{a'}$ le point défini par $y_{(i,j)} = y_{(j,i)} = 0$, $y_a = 1$ pour tout $a \in A_0 \cup A_0^{-1} \setminus \{(i,j),(j,i)\}$. On note par Z_0 le point défini par $y_a = 1$ pour tout arc $a \in A_0 \cup A_0^{-1}$. Il est facile de voir que ces points appartiennent à $P_U(C_n, v_0, K)$. On montre maintenant que tout point $Z = (x,y)$ peut s'écrire comme combinaison convexe des points Z_0 et Z_a, $a \in A_0$, *i.e.*,

$$Z = \sum_{a \in A_0} (1 - y_a) Z_a + (1 - \sum_{a \in A_0} (1 - y_a)) Z_0.$$

En effet, il est facile de voir que la somme des coefficients est égale à un. Il reste maintenant à montrer que les coefficients de cette combinaison sont positifs ou nuls. On suppose, sans perte de généralité, que l'ensemble A_0 est non vide. D'après les contraintes triviales (4.17), les coefficients $(1 - y_a)$ sont positifs ou nuls. Considérons maintenant le coefficient $(1 - \sum_{a \in A_0}(1 - y_a))$. Pour que ce dernier soit positif ou nul, il faut que $\sum_{a \in A_0} y_a \geq |A_0| - 1$. Supprimons dans C_n les arcs de A_0 et A_0^{-1}. Le graphe résultant est un ensemble de composantes fortement connexes. Pour chaque composante, on note par W_i l'ensemble de ses sommets. On obtient alors une partition des sommets $\pi = \{W_1, W_2, \ldots, W_s\}$ de V. Supposons, sans perte de généralité, que v_0 appartient à W_1. Comme on considère les circuits restreints, il est clair que chaque sous-ensemble W_i, $i = 2, 3, \ldots, s$, contient l'extrémité d'une demande. De plus, on a $s = |A_0|$. La contrainte de partition associée à π assure que le dernier coefficient est positif ou nul, ce qui implique que la combinaison est convexe.

On considère maintenant le cas où (C_n, v_0, K) ne peut être réduit à un circuit double origine/destination. Nous allons distinguer deux cas. Supposons d'abord que (C_n, v_0, K) peut être réduit au circuit donné dans la figure 5.2. On suppose, par la suite, que les demandes k et k' correspondent respectivement aux demandes 1 et 2. Par ailleurs, la preuve est donnée uniquement dans le cas où les sommets o^1 et d^2 sont distincts. Les autres cas ne seront pas traités car ils sont similaires ou faciles. Comme (C_n, v_0, K) ne peut être réduit à un circuit double origine/destination, le circuit passe par les sommets v_0, o^1, d^2 et $o^2 = d^1$. Le sous-ensemble V_1 (respectivement V_2) fait maintenant référence aux sommets du chemin de o^1 à d^1 (respectivement o^2 à d^2) contenant (respectivement ne contenant pas) le sommet v_0. Par ailleurs, P_1 et P_2 correspondent respectivement aux chemins de o^1 à d^2 et de o^2 à d^1 ne passant pas par le sommet v_0. (Si $o^2 = d^1$, alors P_2 est vide.) Comme (C_n, v_0, K) ne contient pas de circuit double origine/destination, il est facile de voir que toutes les demandes de $K \setminus \{1,2\}$ sont de type (v_i, v_j) avec $i > j$, et v_i et v_j appartiennent tous les deux à P_1 ou à P_2.

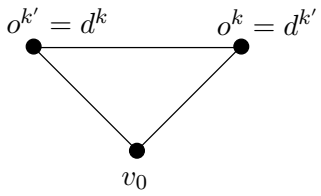

FIGURE 5.2 – Circuit à 3 sommets

Comme (C_n, v_0, K) ne peut être réduit à un circuit double origine/destination, le sommet v_0 n'est incident à aucune demande. Pour chaque demande $k \in K$, on note par L_k^1 (respectivement L_k^2) le $o^k d^k$-chemin élémentaire passant (respectivement ne passant pas) par v_0. Toute solution (x, y) de $P_U(C_n, v_0, K)$ vérifie alors

$$x_a^k = \begin{cases} 1 & \text{si } a \in L_k^2, \\ 0 & \text{sinon,} \end{cases} \qquad \forall\ k \in K \setminus \{1\}. \tag{5.127}$$

En effet, supposons qu'il existe $k \in K \setminus \{1\}$ telle que $x_a^k = 1$ pour tout arc $a \in L_k^1$. Comme ce chemin intersecte P_1, les contraintes de capacité impliquent que la demande 1 est transportée sur L_1^1. Les contraintes de capacité impliquent que l'on a $x^1(\delta^{\text{out}}(v_0)) + x^k(\delta^{\text{out}}(v_0)) = 2$. De plus, on a $x^1(\delta^{\text{in}}(v_0)) + x^k(\delta^{\text{in}}(v_0)) = 2$. Comme $\deg^{\text{out}}(v_0) = 2$ et $\deg^{\text{in}}(v_0) = 2$, si v_0 est incident à une demande, alors cette demande ne peut être transportée de son origine à sa destination sans intersecter les chemins des demandes 1 et k. Il existe donc une contrainte de capacité associée à un arc de $\delta(v_0)$ qui est violée par (x, y). Si v_0 n'est incident à aucune demande, la contrainte de vulnérabilité renforcée (4.24) associée à v_0 est violée par (x, y).

On suppose, sans perte de généralité, que (x, y) satisfait

$$x_a(K^a \setminus \{1\}) \leq \begin{cases} 0 & \text{si } a \in L_1^1 \cup L_1^2, \\ 1 & \text{sinon,} \end{cases} \qquad \forall\ a \in A.$$

En effet, si (x, y) ne vérifie pas ces équations, alors soit $P_U(C_n, v_0, K)$ est vide, soit la demande 1 ne peut être transportée que sur un des deux chemins et dans ce cas, x est fixé et la preuve est la même que précédemment.

Il est clair que tout point (x, y) de $P_U(C_n, v_0, K)$ satisfait les équations $y_{(i,j)} = y_{(j,i)}$ pour tout $(i, j) \in A$. En effet, si la demande 1 est transportée sur L_1^1, alors on a $x^1_{(v_0, v_{n-1})} = 1$, ce qui implique, d'après la contrainte de vulnérabilité renforcée (4.24) associée à v_0, que $y(\delta^{\text{out}}(v_0)) = 2$. On a alors $y_{(v_0,v_1)} = 1$. La contrainte de conservation de flot (4.11) associée à v_0 implique que $y_{(v_1,v_0)} = 1$. Si la demande 1 est transportée sur L_1^2, il existe alors, par les contraintes de capacité, un arc (i, j), $i, j \in V_2$, tel que

$y_{(i,j)} = y_{(j,i)}$. Les contraintes de conservation de flot assurent alors que les équations sont satisfaites pour tout arc de A.

Notons respectivement A_1 et A_2 les ensembles d'arcs des chemins L_1^1 et L_1^2 tels qu'aucune demande de $K \setminus \{1\}$ n'est transportée ni sur ces arcs, ni sur leur arcs inverses, c'est-à-dire, $A_k = \{(i,j) \in L_1^k : x_{(i,j)}(K^{(i,j)} \setminus \{1\}) = x_{(j,i)}(K^{(j,i)} \setminus \{1\}) = 0\}$, $k = 1, 2$.

Soit a_1 un arc de L_1^1. Tout point $Z = (x, y)$ n'est identifié que si les composantes x_a^1 et y_a, $a \in A_1 \cup A_2$ sont définies. En effet, d'après (5.127), x^k est défini pour tout $k \neq 1$. Si x_a^1 est défini, alors par les contraintes de conservation de flot (4.12) associées à la demande 1, le vecteur x^1 est défini. Par ailleurs, comme $y_{(i,j)} = y_{(j,i)}$ pour tout $(i,j) \in A$, les contraintes de capacité (4.15) impliquent que pour tout arc $a \in A \setminus (A_1 \cup A_2)$, on a $y_a = 1$. Le vecteur y n'est défini que si les valeurs y_a, $a \in A_1 \cup A_2$ le sont. Par conséquent, toute solution $Z = (x, y)$ sera seulement donnée par les composantes y_a, $a \in A_1 \cup A_2$.

Pour tout arc $a' = (i,j) \in A_1$, on note par $Z_{a'} = (x, y)$ le point tel que $y_{(i,j)} = y_{(j,i)} = 0$, $y_a = 1$ pour tout $a \in A_1 \cup A_2 \setminus \{(i,j),(j,i)\}$ et $x_{a_1}^1 = 0$. De la même manière, pour tout arc $a' = (i,j) \in A^2$, $Z_{a'}$ est le point tel que $y_{(i,j)} = y_{(j,i)} = 0$, $y_a = 1$ pour tout $a \in A_1 \cup A_2 \setminus \{(i,j),(j,i)\}$ et $x_{a_1}^1 = 1$. On considère également les points Z^1 et Z^2 définis par $y_a = 1$ pour tout arc $a \in A_1 \cup A_2$ et par $x_{a_1}^1 = 0$ et $x_{a_1}^1 = 1$ respectivement. Il est facile de voir que ces points appartiennent à $P_U(C_n, v_0, K)$. On montre maintenant que tout point $Z = (x, y)$ s'écrit comme combinaison convexe des points Z^1, Z^2 et Z_a, $a \in A_1 \cup A_2$, *i.e.*,

$$Z = \sum_{a \in A_1 \cup A_2} (1 - y_a) Z_a + ((1 - x_{a_1}^1) - \sum_{a \in A_1} (1 - y_a)) Z^1 + (x_{a_1}^1 - \sum_{a \in A_2} (1 - y_a)) Z^2.$$

En effet, la somme des coefficients est égale à un. Il suffit maintenant de montrer que les coefficients de cette combinaison sont positifs ou nuls. Il est clair que ceci est vrai pour les coefficients $(1 - y_a)$ par les contraintes triviales (4.17).
Considérons maintenant le coefficient associé au point Z^1. On suppose, sans perte de généralité, que A_1 est non vide. Supposons que A_1 contient un seul arc, disons $\bar{a}$. D'après la définition de l'ensemble A^1 associé à la demande 1 et les contraintes de conservation de flot (4.12), on déduit que $x_{\bar{a}}^1 = x_{a_1}^1$. La contrainte de capacité (4.15) associée à l'arc $\bar{a}$ implique que $y_{\bar{a}} \geq x_{\bar{a}}^1$. Le coefficient est donc non-négatif. Supposons maintenant que A_1 contient au moins deux arcs. Soit π la partition obtenue par suppression des arcs (i,j) tels que (i,j) ou (j,i) appartient à A_1. Le nombre de sous-ensembles de π est égal à $|A_1|$. Considérons la contrainte de partition associée à π. On a alors $y(A_\pi) - x_{a_0}(K_\pi) \geq |A_1| - 1$. De plus, pour tout arc (i,j) appartenant à A_π, (i,j) ou (j,i) appartient à A_1. On a alors $y(A_\pi) = y(A_1)$. Par ailleurs, les sommets du chemin

L_2 appartiennent à un même sous-ensemble de la partition π, ce qui implique que la demande 1 a ses deux extrémités dans un sous-ensemble de π. De plus, l'arc a_0 associé à la partition π appartient au chemin L_1^1. Par conséquent, la demande 1 appartient à K_π. La contrainte de partition associée à π assure alors que $y(A_1) - x^1_{a_1} \geq |A_1| - 1$. Le coefficient associé à Z^1 est donc positif ou nul. On montre de la même manière que le coefficient associé à Z^2 est non-négatif. Il en résulte que Z est une combinaison convexe de points de $P_U(C_n, v_0, K)$.

Supposons maintenant que (C_n, v_0, K) ne peut être réduit au circuit à 3 sommets de la figure 5.2. Dans ce cas, toute demande $k \in K$ est telle que l'arc (o^k, d^k) appartient à A^+. Comme (C_n, v_0, K) ne contient pas de sommet isolé, on suppose, sans perte de généralité, que le circuit passe successivement par les sommets v_0, o^k, d^k, $k \in K$. Posons $\alpha = y_{(v_0,v_1)} - y_{(v_1,v_0)}$. D'après les contraintes de conservation de flot associées à y, on obtient $y_{(i,j)} = y_{(j,i)} - \alpha$ pour tout $(i,j) \in A^-$. Pour toute demande $k \in K$, les contraintes de conservation de flot (4.12) implique que l'on a $x^k_a = 1 - x^k_{(o^k,d^k)}$ pour tout $a \in A^k \setminus \{(o^k, d^k)\}$. Tout point $Z = (x, y)$ sera seulement donné par les composantes $x_{(o^k,d^k)}$, $k \in K$, α et y_a, $a \in A^+$. La composante $y_{(i,j)}$, $a \in A^-$, est donnée par $y_{(j,i)}$ et α.

Considérons les points de $P_U(C_n, v_0, K)$ définis de la manière suivante. Pour tout $a' \in A^+$, on note par $Z_{a'}$ le point défini par $y_{a'} = 0$, $y_a = 1$ pour tout arc $a \in A^+ \setminus \{a'\}$, $\alpha = 0$, et

$$x^k_{(o^k,d^k)} = \begin{cases} 1 & \text{si } a' \neq (o^k, d^k), \\ 0 & \text{sinon,} \end{cases} \qquad \forall\ k \in K.$$

Considérons un deuxième ensemble de points comme suit. Soit k' une demande de K. Notons par $Z_{k'}$ le point défini par $y_a = 1$ pour tout $a \in A^+$, $\alpha = 0$, $x^{k'}_{(o^{k'},d^{k'})} = 0$ et $x^k_{(o^k,d^k)} = 1$ pour tout $k \neq k'$. Finalement, notons par Z^1 (respectivement Z^2) le point défini par $y_a = 1$ pour tout $a \in A^+$, $x^k_{(o^k,d^k)} = 1$ pour tout $k \in K$ et $\alpha = 1$ (respectivement $\alpha = 0$). Il est facile de voir que ces points appartiennent à $P_U(C_n, v_0, K)$. Soit $A' = A^+ \setminus \{(o^k, d^k) : k \in K\}$. Tout point $Z = (x, y)$ peut alors s'écrire comme combinaison convexe des points Z^1, Z^2, Z_a, $a \in A^+$, et Z_k, $k \in K$, *i.e.*,

$$\begin{aligned} Z = \ & \textstyle\sum_{a\in A^+}(1 - y_a)Z_a + \sum_{k\in K}(y_{(o^k,d^k)} - x^k_{(o^k,d^k)})Z_k + \alpha Z^1 \\ & + (1 - |A^+| + \textstyle\sum_{k\in K} x^k_{(o^k,d^k)} + \sum_{a\in A'} y_a - \alpha)Z^2 \end{aligned}$$

On montre d'abord que les coefficients sont positifs ou nuls. Ceci est vérifié pour les coefficients associés aux points Z_a, $a \in A^+$ grâce aux contraintes triviales. Considérons les coefficients associés aux points Z_k, $k \in K$. Ces coefficients sont positifs ou nuls car les contraintes de capacité impliquent que $y_{(o^k,d^k)} \geq x^k_{(o^k,d^k)}$. La contrainte de priorité

(5.120) assure que α est positif ou nul. Considérons maintenant le coefficient associé au point Z^2. Considérons la partition de sommets $\pi = \{W_1, W_2, \ldots, W_{p+1}\}$ avec $W_1 = \{v_0\}$, $W_{k+1} = \{o^k, d^k\}$, $k = 1, 2, \ldots, p$. Il est facile de voir que l'ensemble A_π est équivalent à l'ensemble $A' \cup \{(v_0, d^p)\} \setminus \{(d^p, v_0)\}$. La contrainte de partition associée à π implique que $y(A_\pi) - \sum_{k \in K}(1 - x^k_{(o^k, d^k)}) \geq p$. De plus, on a $y(A_\pi) = y(A') - \alpha$ et $|A^+| = 2p + 1$, ce qui implique que le coefficient associé à Z^2 est positif ou nul. Finalement, il est facile de voir que la somme des coefficients est égale à 1, ce qui implique que Z est une combinaison convexe. □

Dans la section suivante, nous étendons ce résultat aux cactus si l'ensemble des demandes K vérifie certaines hypothèses.

5.3.4 Composition de polyèdres

Soit $D = (V, A)$ un graphe orienté. On dit que D est la *1-somme* de $D^1 = (V_1, A_1)$ et $D^2 = (V_2, A_2)$ si $V = V_1 \cup V_2$, $|V_1 \cap V_2| = 1$, $A = A_1 \cup A_2$ et $A_1 \cap A_2 = \emptyset$. (Voir figure 5.3.)

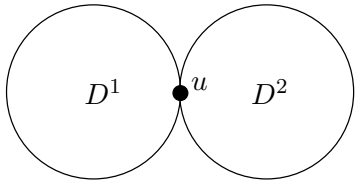

FIGURE 5.3 – 1-somme de D^1 et D^2

Dans cette section, nous allons décrire le polytope $P_U(D, v_0, K)$ dans le cas où D est la 1-somme de deux graphes D^1 et D^2. Soit (D, v_0, K) une instance du 1-PRLPA unitaire. Supposons que $D = (V, A)$ est la 1-somme de $D^1 = (V_1, A_1)$ et $D^2 = (V_2, A_2)$. On note par u le sommet de $V_1 \cap V_2$. Sans perte de généralité, on suppose que le dépôt v_0 est un sommet de V_1 et que toute demande (o^k, d^k) de K est contenue soit dans D^1, soit dans D^2. En effet, si, par exemple, il existe une demande (o^k, d^k) telle que $o^k \in V_1$, $d^k \in V_2$ et $o^k \neq u \neq d^k$, alors on peut remplacer la demande (o^k, d^k) par les deux demandes (o^{k_1}, d^{k_1}) et (o^{k_2}, d^{k_2}) telles que $o^{k_1} = o^k$, $d^{k_1} = o^{k^2} = u$ et $d^{k_2} = d^k$.

Notons par K^1 et K^2 les demandes ayant leurs deux extrémités dans D^1 et D^2 respectivement. Supposons que $u = v_0$ ou u est incident à au moins une demande de K^1.

Dans la suite, nous allons montrer que, sous ces hypothèses, le polytope $P_U(D, v_0, K)$ peut être décrit à partir des polytopes $P_U(D^1, v_0, K^1)$ et $P_U(D^2, u, K^2)$. Soit

$$P_0 = \begin{cases} P_U(D^1, v_0, K^1), \\ P_U(D^2, u, K^2), \\ x_a^k = 0 \qquad\qquad \forall\ k \in K^i, i = 1, 2, \forall\ a \in A^k \setminus A_i. \end{cases} \tag{5.128}$$

Proposition 5.19 *Les contraintes de P_0 sont valides pour $P_U(D, v_0, K)$.*

Preuve. Soit (x, y) une solution de $P_U(D, v_0, K)$. Considérons une demande $k \in K^i$, $i \in \{1, 2\}$. Les sommets o^k et d^k appartiennent alors à V_i. Comme chaque demande doit être transportée sur un chemin élémentaire, pour tout arc $a \in A^k \setminus A_i$, on a $x_a^k = 0$. Les contraintes (5.128) sont donc valides pour $P_U(D, v_0^1, K)$.
Soit (x^1, y^1) la restriction de (x, y) sur (D^1, v_0, K^1). On montre que (x^1, y^1) appartient à $P_U(D^1, v_0, K^1)$. Il est clair que (x^1, y^1) vérifie les contraintes triviales (4.16)-(4.19). Supposons que (x^1, y^1) viole une contrainte de conservation de flot (4.11) associée à un sommet $v \in V^1 \setminus \{u\}$. Dans ce cas, la contrainte est également violée par (x, y), ce qui contredit l'hypothèse que (x, y) est une solution de $P_U(D, v_0, K)$. Les contraintes (4.11) sont donc satisfaites pour tout sommet $v \in V^1 \setminus \{u\}$. On déduit alors que $y^1(\delta^{\text{out}}(V^1 \setminus \{u\})) - y^1(\delta^{\text{in}}(V^1 \setminus \{u\})) = 0$. Il s'ensuit que $y^1(\delta^{\text{out}}_{D^1}(u)) - y^1(\delta^{\text{in}}_{D^1}(u)) = 0$. Le point (x^1, y^1) vérifie donc les contraintes de conservation de flot (4.11).
Considérons une demande $k \in K^1$. Pour tout sommet $v \in V^1 \setminus \{u\}$, la contrainte de conservation de flot (4.12) associée et la contrainte de circuit (4.13) associées à k et v sont vérifiées par (x^1, y^1) puisque (x, y) satisfait ces contraintes. Considérons les contraintes (4.12) et (4.13) associées à k et u. Par les contraintes (5.128), on a $x^k(\delta_{D^2}(u)) = 0$ pour tout $k \in K^1$. On déduit alors que $x^k(\delta(u)) = x^k(\delta_{D^1}(u)) = 0$, ce qui implique que les contraintes (4.12) et (4.13) sont satisfaites par (x^1, y^1). Il est facile de voir, par un raisonnement similaire, que (x^1, y^1) vérifie aussi les contraintes de connexité des demandes (4.14). De plus, comme $K^1 \subseteq K$, les contraintes de capacité (4.15) sont également vérifiées par (x^1, y^1).
Comme (x, y) appartient à $P_U(D, v_0, K)$, il vérifie donc les contraintes de connexité (4.10) et les contraintes de vulnérabilité renforcées (4.24) associées aux sous-ensembles $W \subseteq V_1$ (et aux arcs $a \in A_1$ pour les contraintes (4.10)). Celles-ci sont donc vérifiées par (x^1, y^1), ce qui implique que ce dernier appartient à $P_U(D^1, v_0, K^1)$. Les contraintes de $P_U(D^1, v_0, K^1)$ sont donc valides pour $P_U(D, v_0, K)$.
On peut montrer, de manière similaire, que les contraintes de $P_U(D^2, u, K^2)$ sont valides pour $P_U(D, v_0, K)$. On a alors $P_U(D, v_0, K) \subseteq P_0$. □

Proposition 5.20 *Si $u = v_0$ ou u est incident à au moins une demande de K^1, alors toute solution entière (x, y) de P_0 appartient à $P_U(D, v_0, K)$.*

Preuve. Considérons une solution entière (x, y) de P_0. Comme $A = A_1 \cup A_2$ et $K = K^1 \cup K^2$, (x, y) vérifie les contraintes triviales (4.16)-(4.19). De plus, (x, y) vérifie les contraintes de conservation de flot (4.11) associées aux sommets de $V \setminus \{u\}$. De plus, on a $y(\delta^{\text{out}}_{D^i}(u)) - y(\delta^{\text{in}}_{D^i}(u)) = 0$ pour $i = 1, 2$. Comme $\delta^{\text{out}}_D(u) = \delta^{\text{out}}_{D^1}(u) \cup \delta^{\text{out}}_{D^2}(u)$ et $\delta^{\text{in}}_D(u) = \delta^{\text{in}}_{D^1}(u) \cup \delta^{\text{in}}_{D^2}(u)$, (x, y) vérifie la contrainte de conservation de flot (4.11) associée à u.
Comme (x, y) vérifie les équations (5.128), il est facile de voir que (x, y) vérifie les contraintes (4.12)-(4.13). Soit a un arc de A_i, $i = 1, 2$. Comme (x, y) appartient à P_0, il vérifie $y_a - \sum_{k \in K^i \cap K^a} x^k_a \geq 0$. Comme $x^k_a = 0$ pour tout $k \in K \setminus K^i$, alors $y_a - \sum_{k \in K^a} x^k_a \geq 0$. Les contraintes de capacité (4.15) sont donc vérifiées par (x, y).
Supposons maintenant que (x, y) ne vérifie pas une contrainte de connexité (4.10). Comme (x, y) est entier, cela implique que le graphe D_y induit par l'ensemble d'arcs $\{a \in A : y_a = 1\}$ n'est pas connexe. Si D_y contient un arc a appartenant à $A_2 \setminus \delta(u)$, alors, par la contrainte de connexité (4.10) associée à $\{u\}$ et a, il s'ensuit que $y(\delta^{\text{out}}_{D^2}(u)) \geq 1$. De même, si $u = v_0$, alors on a $y(\delta^{\text{in}}_{D^1}(u)) \geq 1$. Supposons maintenant que $u \neq v_0$. Par hypothèse, il existe une demande de K^1 incidente à u. On a alors $\sum_{k \in K^1} x^k(\delta^{\text{in}}_{D^1}(u)) \geq 1$. Les contraintes de capacité impliquent que $y(\delta^{\text{in}}_{D^1}(u)) \geq 1$. Considérons les graphes D^1_y et D^2_y correspondant respectivement aux restrictions de D_y sur D^1 et D^2. Si D_y n'est pas connexe, alors D^1_y ou D^2_y ne l'est pas non plus. Ceci implique qu'il existe une contrainte de connexité dans D^1 ou D^2 qui est violée par (x, y), contredisant alors l'hypothèse que (x, y) est un point de P_0. Les contraintes de connexité (4.10) sont donc vérifiées par (x, y).
Supposons qu'il existe une contrainte de vulnérabilité renforcée (4.24), disons celle associée à l'ensemble $W \subseteq V$, qui est violée par (x, y). Pour tout $k \in K^i$, on a $x^k(\delta^{\text{out}}_{D^i}(W) \cap A^k) = x^k(\delta^{\text{out}}(W) \cap A^k)$. Soient W^i, $i = 1, 2$, la restriction de W sur D^i. La contrainte de vulnérabilité renforcée associée à W n'est rien d'autre que la somme des contraintes de vulnérabilité renforcées associées à W^1 et W^2 dans D^1 et D^2 respectivement. Le point (x, y) vérifie alors les contraintes de vulnérabilité renforcée (4.24). On déduit alors que (x, y) appartient à $P_U(D, v_0, K)$. □

Théorème 5.21 *Si $u = v_0$ ou u est incident à au moins une demande de K^1, alors $P_U(D, v_0, K) = \mathcal{P}_0$.*

Preuve. Par la proposition 5.19, nous avons $P_U(D, v_0, K) \subseteq P_O$ et par la proposition 5.20, toute solution entière de P_0 appartient à $P_U(D, v_0, K)$. Il reste à prouver

que tout point extrême de P_0 est entier. Supposons qu'il existe un point extrême fractionnaire (x, y) de P_0 et soit S le système de contraintes de P_0 vérifiées à l'égalité par (x, y). Il est clair que les équations (5.128) appartiennent à S. Soient (x^1, y^1) et (x^2, y^2) les restrictions de (x, y) sur (D^1, v_0, K^1) et (D^2, u, K^2) respectivement. Notons par S^1 (respectivement S_2) le système d'équations de $P_U(D^1, v_0, K^1)$ (respectivement $P_U(D^1, u, K^1)$) apparaissant dans S. Supposons que (x^1, y^1) est fractionnaire. Comme $P_U(D^1, v_0, K^1)$ est un polyèdre entier, alors il existe un point entier de $P_U(D^1, v_0, K^1)$, disons $(\bar{x}, \bar{y})$ vérifiant S^1. De même, si (x^2, y^2) est fractionnaire, alors il existe un point entier $(\tilde{x}, \tilde{y})$ de $P_U(D^2, u, K^2)$ vérifiant S^2. Considérons la solution $(\hat{x}, \hat{y})$ définie par

$$\hat{x}_a^k = \begin{cases} \bar{x}_a^k & \text{si } a \in A_1 \text{ et } k \in K^1, \\ \tilde{x}_a^k & \text{si } a \in A_2 \text{ et } k \in K^2, \\ 0 & \text{sinon,} \end{cases} \qquad \forall\, k \in K, \forall\, a \in A^k,$$

et

$$\hat{y}_a = \begin{cases} \bar{y}_a & \text{si } a \in A_1, \\ \tilde{y}_a & \text{sinon,} \end{cases} \qquad \forall\, a \in A.$$

Le point $(\hat{x}, \hat{y})$ vérifie les systèmes S^1, S^2 et les contraintes (5.128), ce qui implique qu'il vérifie S. De plus, il est clair qu'il appartient à P_0. Comme $(x, y) \neq (\hat{x}, \hat{y})$, ceci contredit le fait que (x, y) est un point extrême de P_0. Le polytope P_0 est donc entier. □

5.3.5 Application aux cactus

Comme application des résultats de la section précédente, nous allons étudier le polytope dans la classe des cactus formés par des circuits doublement orientés. Soit (D, v_0, K) une instance du 1-PRLPA unitaire. Supposons que D est un cactus formé par les circuits doublement orientés $C_1, C_2, \ldots, C_l$, avec $C_i = (V_i, A_i)$, $i = 1, 2, \ldots, l$. Sans perte de généralité, on suppose que toute demande $k \in K$ a ses deux extrémités dans un même circuit C_i, $i = 1, 2, \ldots, l$. Dans le cas contraire, on peut appliquer la procédure suivante jusqu'à ce que toute les demandes aient leur deux extrémités dans un même circuit :

– Soit $k \in K$ une demande n'ayant pas ses deux extrémités dans un même circuit. Notons par C_i et C_j les circuits contenant respectivement l'origine et la destination de la demande k. Remplacer la demande k dans K par deux demandes k_1 et k_2 telles que $o^{k_1} = o^k$, $d^{k_2} = d^k$ et $d^{k_1} = o^{k_2} = v$, où v est le sommet d'articulation de C_i tel que o^k et d^k appartiennent à deux composantes connexes différentes dans $D \setminus v$.

On note par K_i , $i = 1, 2, \ldots, l$, les demandes de K ayant leur deux extrémités dans le circuit C_i. Alors $K = \cup_{i=1}^{l} K_i$. On associe à chaque circuit C_i, $i = 1, 2, \ldots, l$, un sommet v_i de la manière suivante. Si $v_0 \in C_i$, alors $v_i = v_0$. Sinon, le sommet v_i correspond au sommet d'articulation de C_i tel que le graphe $D \setminus v_i$ contient deux composantes connexes, l'une contenant les arcs de $C_i \setminus \delta(v_i)$, l'autre contenant le sommet v_0. À partir du théorème 5.21, nous déduisons le corollaire suivant.

Corollaire 5.22 *Si chaque sommet différent de v_0 est incident à au moins une demande et si, pour tout sommet d'articulation v différent de v_0, il existe une demande de K ayant comme extrémités v et un sommet de la composante connexe contenant v_0 dans $D \setminus v$, alors $P_U(D, v_0, K)$ est donné par*

$$P_U(D, v_0, K) = \begin{cases} P_U(C_i, v_i, K_i) & \forall\ i = 1, 2, \ldots, l, \\ x_a^k = 0 & \forall\ k \in K_i, i = 1, 2, \ldots, l, \forall\ a \in A^k \setminus A_i. \end{cases}$$

□

5.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés au polytope des solutions du 1-PRLPA unitaire. Nous avons caractérisé sa dimension puis nous avons donné des conditions nécéssaires et suffisantes pour que les contraintes de la formulation (4.23) définissent des facettes. En utilisant ces résultats, nous avons développé un algorithme de coupes et branchements permettant de résoudre le 1-PRLPA unitaire. Les résultats expérimentaux ont montré la qualité de la formulation (4.23). L'algorithme de coupes et branchements a permis de résoudre efficacement des instances de taille moyenne. De plus, le nombre de nœuds dans l'arbre de branchements était généralement très faible. Pour certaines instances, lorsque l'algorithme n'a pas trouvé une solution optimale dans le temps imparti, une solution proche de l'optimum a été trouvée. L'erreur relative entre cette solution et la borne inférieure était toujours très faible, ce qui montre que la formulation est très serrée. Cependant, la résolution de la relaxation linéaire initiale prend beaucoup de temps.
Nous avons aussi caractérisé le polytope des solutions du 1-PRLPA unitaire lorsque le graphe est un cactus, chaque sommet différent du dépôt est incident à au moins une demande et pour tout sommet d'articulation $v \neq v_0$, il existe, dans le graphe induit par $D \setminus v$, une demande ayant comme extrémités un sommet de la composante connexe contenant v_0 et le sommet v.

Un travail de recherche serait nécessaire pour améliorer la résolution du programme linéaire initial (5.118). Des pré-traitements pourraient être envisagés afin de réduire l'ensemble des solutions. De même, il serait intéressant de développer un algorithme de résolution basé sur la génération de colonnes. En effet, actuellement, le nombre de variables du problème devient rapidement très grand même pour des instances de taille moyenne. Un algorithme de génération de colonnes renforcé par une procédure de coupes pourrait donner de meilleurs résultats.
Concernant la description polyédrale, il serait intéressant d'étendre la description du polytope $P_U(D, v_0, K)$ dans des circuits arbitraires. Dans la composition polyédrale, l'hypothèse que le sommet d'articulation u soit égal à v_0 ou soit incident à une demande de K^1 (sachant que le dépôt appartient à V_1) est assez restrictive. Il serait bon d'étendre la composition polyédrale au cas général où cette hypothèse n'est pas considérée. Enfin, il serait intéressant d'étendre cette étude au cas de la 2-somme. (Un graphe D est la *2-somme* de deux graphes D_1 et D_2 si D peut être obtenu à partir de D_1 et D_2 par l'identification d'un arc.)

n	p	opt	N_{CD}	N_{Vuln}	N_T	Gap	T_{CD}	T_{Vuln}	TT
31	5	5/5	0.00	25.60	0.00	0.10	0.00	0.02	0.53
31	10	5/5	0.00	28.20	1.60	0.18	0.01	0.09	1.38
31	15	5/5	0.00	3.60	0.00	0.00	0.00	0.02	1.59
41	5	5/5	10.20	42.80	6.00	0.27	0.03	0.12	5.04
41	10	5/5	0.00	31.60	0.00	0.00	0.02	0.07	3.45
41	15	5/5	0.00	53.80	0.80	0.00	0.04	0.49	8.70
41	20	5/5	0.00	12.40	3.60	0.00	0.01	0.28	10.28
51	5	5/5	20.00	108.80	0.00	0.03	0.09	0.61	15.50
51	10	5/5	0.00	85.80	0.40	0.03	0.06	0.76	41.17
51	15	5/5	0.00	33.40	3.60	0.01	0.05	0.18	27.63
51	20	5/5	0.00	49.60	2.60	0.00	0.06	1.05	36.44
51	25	5/5	0.00	8.60	0.40	0.00	0.01	0.39	41.80
61	5	5/5	0.00	27.00	0.00	0.02	0.01	0.10	7.24
61	10	5/5	0.00	34.60	7.80	0.04	0.05	0.47	27.64
61	15	5/5	0.00	147.80	0.00	0.01	0.23	2.79	178.35
61	20	5/5	0.00	72.20	1.60	0.01	0.12	1.51	125.88
61	25	5/5	0.00	63.40	6.20	0.02	0.18	2.43	296.22
61	30	5/5	0.00	7.80	1.00	0.00	0.02	0.65	166.55
71	5	5/5	9.60	433.40	7.20	0.03	0.44	4.81	134.97
71	10	5/5	0.00	124.20	0.00	0.01	0.23	2.54	80.20
71	15	4/5	0.00	515.60	2.00	0.06	0.63	15.75	1513.00
71	20	5/5	0.00	62.80	0.20	0.00	0.24	1.13	271.85
71	25	5/5	0.00	104.80	6.80	0.00	0.46	6.53	382.85
71	30	5/5	0.00	26.40	4.20	0.00	0.13	1.65	490.17
71	35	5/5	0.00	17.20	0.00	0.00	0.08	1.87	646.57
81	5	5/5	0.00	75.60	1.20	0.01	0.09	0.42	19.60
81	10	5/5	2.80	195.60	3.60	0.01	0.54	3.71	127.45
81	15	5/5	0.00	90.20	12.80	0.01	0.60	5.82	232.71
81	20	5/5	0.00	217.20	5.80	0.00	0.87	7.31	472.35
81	25	5/5	4.00	213.20	2.40	0.00	0.80	9.16	637.40
81	30	5/5	0.00	164.40	9.60	0.00	0.89	10.77	916.14
81	35	5/5	0.00	68.60	0.00	0.00	0.40	1.79	1228.94
81	40	5/5	0.00	16.60	0.80	0.00	0.13	4.25	1632.07
91	5	5/5	23.20	647.20	39.80	0.04	1.19	13.89	343.95
91	10	5/5	4.00	640.60	98.80	0.02	2.40	31.81	503.49
91	15	5/5	44.00	654.20	38.40	0.03	4.14	48.51	1264.06
91	20	5/5	0.00	278.00	9.20	0.01	1.87	15.63	826.81
91	25	4/5	12.40	682.40	14.00	0.01	3.50	44.04	2408.21
91	30	5/5	0.00	196.80	9.00	0.00	2.08	19.28	2092.57
91	35	5/5	0.00	58.40	1.20	0.00	0.40	6.09	2187.17
91	40	5/5	0.00	77.80	4.00	0.00	0.70	16.86	3186.65
91	45	5/5	0.00	9.40	1.60	0.00	0.13	3.64	3857.57
101	5	5/5	11.20	557.00	1.20	0.03	0.84	12.04	323.83
101	10	5/5	4.40	191.00	4.40	0.00	1.08	5.80	425.82
101	15	5/5	0.00	296.20	0.00	0.00	1.87	17.04	992.85
101	20	5/5	0.00	640.00	13.80	0.01	6.32	109.52	2631.06
101	25	5/5	0.00	265.60	18.80	0.01	2.69	21.32	2815.44
101	30	5/5	0.00	287.80	12.40	0.00	4.03	38.12	3395.71
101	35	5/5	0.00	167.20	11.40	0.00	2.49	26.27	4465.60
101	40	5/5	0.00	66.40	6.00	0.00	0.99	5.11	5370.81
101	45	4/5	0.00	44.20	3.40	0.19	0.56	2.87	6844.43
101	50	0/5	0.00	0.00	0.00	-	0.00	0.00	7200.00

TABLE 5.1 – Résultats obtenus pour les instances symétriques

n	p	opt	N_{CD}	N_{Vuln}	N_T	Gap	T_{CD}	T_{Vuln}	TT
31	5	5/5	0.00	20.00	0.00	0.00	0.00	0.02	0.44
31	10	5/5	0.00	15.60	0.60	0.01	0.00	0.03	1.13
31	15	5/5	0.00	9.20	2.40	0.15	0.01	0.07	2.21
41	5	5/5	12.00	55.60	5.60	0.01	0.03	0.09	5.71
41	10	5/5	30.60	125.80	0.00	0.00	0.09	0.90	71.30
41	15	5/5	2.40	52.00	0.20	0.00	0.03	0.48	34.22
41	20	5/5	0.00	4.20	0.60	0.00	0.00	0.09	8.89
51	5	5/5	0.00	20.80	0.00	0.00	0.01	0.06	2.05
51	10	5/5	0.00	11.40	0.60	0.00	0.01	0.10	6.55
51	15	5/5	0.00	97.40	2.20	0.02	0.08	1.04	27.94
51	20	5/5	0.00	33.60	2.40	0.15	0.05	0.74	41.60
51	25	5/5	0.00	4.40	0.00	0.00	0.01	0.24	36.30
61	5	5/5	0.00	43.40	0.00	0.00	0.02	0.30	7.97
61	10	5/5	0.00	19.20	0.00	0.00	0.01	0.20	14.68
61	15	5/5	0.00	205.00	0.20	0.00	0.24	3.81	161.83
61	20	5/5	0.00	14.80	0.60	0.00	0.02	0.58	59.28
61	25	5/5	0.00	44.20	2.00	0.00	0.08	1.98	170.57
61	30	5/5	0.00	5.00	0.40	0.00	0.02	0.36	127.52
71	5	5/5	0.00	1.40	0.00	0.25	0.00	0.00	0.55
71	10	5/5	0.00	9.60	0.80	0.00	0.01	0.09	86.87
71	15	5/5	0.00	10.60	0.00	0.04	0.01	0.29	32.50
71	20	5/5	0.00	4.20	1.20	0.05	0.01	0.21	417.93
71	25	5/5	0.00	5.80	0.80	0.11	0.02	0.22	274.82
71	30	5/5	0.00	13.00	1.00	0.06	0.03	1.20	686.09
71	35	5/5	0.00	8.20	0.40	0.02	0.03	0.83	259.59
81	5	5/5	123.00	1140.20	421.00	0.23	6.70	97.36	1769.63
81	10	5/5	0.00	121.80	0.40	0.00	0.22	2.43	176.96
81	15	4/5	0.00	276.00	0.00	0.42	0.76	17.99	1561.71
81	20	5/5	0.00	219.80	1.40	0.00	0.51	12.55	1458.39
81	25	4/5	0.00	112.20	1.20	0.92	0.62	7.04	1931.71
81	30	5/5	0.00	16.00	2.40	0.00	0.09	1.77	845.37
81	35	5/5	0.00	9.60	2.80	0.00	0.07	1.27	1149.23
81	40	5/5	0.00	13.40	1.80	0.00	0.09	2.90	1404.16
91	5	5/5	0.00	152.40	0.80	0.01	0.21	1.37	72.18
91	10	5/5	0.00	466.40	2.60	0.01	1.15	12.45	518.38
91	15	5/5	0.00	511.80	1.40	0.05	1.21	17.44	857.56
91	20	5/5	0.00	405.80	0.00	0.00	1.25	22.07	955.36
91	25	5/5	0.00	139.00	1.80	0.00	0.61	9.09	1030.19
91	30	5/5	0.00	307.60	1.60	0.00	1.39	23.28	2086.76
91	35	5/5	0.00	147.80	4.20	0.00	0.76	15.38	2015.45
91	40	5/5	0.00	26.40	4.00	0.00	0.28	5.20	2431.55
91	45	5/5	0.00	8.20	1.80	0.00	0.12	4.72	3130.44
101	5	5/5	0.00	2.20	0.20	0.00	0.00	0.01	2.35
101	10	4/5	15.20	88.40	0.40	0.15	0.17	1.71	1457.59
101	15	5/5	0.00	2.00	0.20	0.00	0.01	0.07	428.35
101	20	5/5	0.00	3.80	0.80	0.00	0.02	0.25	510.40
101	25	5/5	0.00	25.80	0.00	0.05	0.12	3.12	3823.82
101	30	4/5	0.00	1.40	0.20	0.04	0.02	0.09	3228.09
101	35	3/5	0.00	31.80	14.60	0.43	0.35	10.57	3563.17
101	40	5/5	0.00	21.60	0.60	0.00	0.14	3.40	2268.91
101	45	5/5	0.00	16.20	1.00	0.00	0.13	6.65	2857.48
101	50	5/5	0.00	8.00	1.20	0.00	0.12	4.70	4075.84

TABLE 5.2 – Résultats obtenus pour les instances asymétriques

Chapitre 6

Le 1-PRLPA unitaire métrique

Dans ce chapitre, on considère le problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire lorsque les hypothèses suivantes sont vérifiées :

1) le graphe D est complet,
2) chaque sommet du graphe est incident à au plus une demande, le dépôt étant incident à aucune demande,
3) le vecteur coût c associé aux arcs du graphe vérifie les inégalités triangulaires.

Dans ce cas, on parle alors du problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire métrique. Sous ces hypothèses, on montre tout d'abord que l'on peut réduire l'espace des solutions du problème en ne considérant que les solutions vérifiant certaines propriétés. Cette réduction permet alors de décrire toute solution du problème uniquement à l'aide de l'ensemble des arcs traversés par le véhicule. On donne les conditions nécessaires et suffisantes pour qu'une solution soit réalisable, puis on formule le problème à l'aide d'un programme linéaire en nombres entiers. Par la suite, on développe une métaheuristique pour résoudre le problème. On définit pour cela une structure de données permettant de coder efficacement les solutions. On présente ensuite dans la section 6.3.2 une heuristique d'insertion permettant d'obtenir une solution réalisable pour le problème. Cette solution servira, par la suite, de solution initiale pour la métaheuristique. Dans la section 6.3.3, on définit plusieurs opérateurs de voisinage ainsi qu'un algorithme de descente simple permettant d'améliorer la solution initiale. Finalement, on analyse les résultats expérimentaux obtenus par l'heuristique d'insertion et la métaheuristique. Le reste de cette introduction est consacrée à une définition plus formelle du problème et donne clairement les hypothèses considérées dans cette étude.

Le *problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire métrique (1-PRLPA unitaire métrique)* est défini par une instance (D, v_0, K) et un vecteur coût c. Le graphe $D = (V, A)$ représente le réseau, le sommet $v_0 \in V$ correspond au dépôt du véhicule et $K = \{1, 2, \ldots, p\}$ est l'ensemble des demandes. À chaque demande sont associés deux sommets distincts o^k et d^k de V correspondant respectivement à l'origine et la destination de la demande. Le vecteur $c \in \mathbb{R}^A$ est indexé sur les arcs du graphe. Plusieurs hypothèses sont vérifiés par (D, v_0, K) et c. On suppose d'abord que le graphe D est complet. De plus, l'ensemble des demandes K est tel que chaque sommet de $V \setminus \{v_0\}$ est incident à au plus une demande. Le dépôt est incident à aucune demande. Ces deux hypothèses ne sont pas restrictives. Il est en effet possible de modifier toute instance (D, v_0, K) et tout vecteur c en une instance (D', v_0, K') et un vecteur coût c' vérifiant ces hypothèses, de la manière suivante. Si un sommet $v \in V$ est incident à s demandes, $s \geq 2$, alors on considère s copies de ce sommet, c'est-à-dire, on ajoute $s - 1$ nouveaux sommets dans V correspondant à v. Les origines et destinations des s demandes sont alors modifiées afin que chaque copie soit incidente à exactement une demande. Par ailleurs, si $v = v_0$, on ajoute un sommet additionnel qui correspondra au nouveau dépôt et qui ne sera donc incident à aucune demande. Des arcs de coûts nuls sont ajoutés entre tout couple de copies du sommet v. Afin d'obtenir un graphe complet, il est possible d'ajouter des arcs de coût infini.

On suppose également que le vecteur coût satisfait les inégalités triangulaires *i.e.*,

$$c_{(u,v)} + c_{(v,w)} - c_{(u,w)} \geq 0 \qquad \forall\ u \neq v \neq w \neq u \in V.$$

Il est possible de modifier le coût c en $\bar{c}$ afin que ce dernier respecte les inégalités triangulaires. Pour cela, il suffit de fixer le coût $\bar{c}_{uv}$ d'un arc (u, v) comme étant le coût d'un plus court chemin de u à v dans le graphe selon le vecteur coût c. Le vecteur $\bar{c}$ vérifie alors les inégalités triangulaires. Cependant, une telle modification modifie légèrement le problème. En effet, chaque arc (u, v) de coût $\bar{c}_{uv}$ correspond à un plus court chemin selon le coût c dans le graphe D. Une solution optimale par rapport à $\bar{c}$ peut contenir deux arcs, disons (u_1, v_1) et (u_2, v_2), correspondant à deux plus courts chemins passant par un même arc selon le vecteur coût c. Ceci implique que toute solution optimale peut passer plusieurs fois par un même arc dans le graphe D.

Toute instance du 1-PRLPA unitaire vérifiant ces trois hypothèses correspond à une instance du 1-PRLPA unitaire métrique. Par la suite, une instance de ce problème sera notée (D, v_0, K, c). On présente dans la section suivante les solutions optimales de cardinalité minimale et l'on donne quelques propriétés vérifiées par ces solutions.

6.1 Solutions optimales de cardinalité minimale

Avant de déterminer les propriétés structurelles de certaines solutions du 1-PRLPA unitaire métrique, on définit quelques notations. Toute solution de (D, v_0, K, c) est notée par (C, L), où $C = ((v_0, v_1), (v_1, v_2), \ldots, (v_z, v_0))$, $z \geq 1$, est le circuit associé au trajet du véhicule et $L = \{L_1, L_2, \ldots, L_p\}$ est l'ensemble des chemins élémentaires arc-disjoints représentant les trajets des demandes. On dit que (C, L) est une *Solution Optimale de Cardinalité Minimale (SOCM)* s'il n'existe pas de solution optimale utilisant moins d'arcs dans le circuit du véhicule.

Résoudre le 1-PRLPA unitaire métrique consiste à déterminer une solution optimale du problème. On peut alors, sans perte de généralité, limiter l'espace de recherche aux seules solutions optimales de cardinalité minimale. Dans ce qui suit, on donne quelques propriétés des SOCM qui seront utilisées par la suite. Par un souci de clarté, on donne une première proposition technique qui définit certaines propriétés structurelles des solutions optimales de cardinalité minimale.

Proposition 6.1 *Soit (C, L) une solution optimale de cardinalité minimale, et v un sommet de $V \setminus \{v_0\}$. Les propriétés suivantes sont vraies.*

I - Le dépôt v_0 apparaît dans C seulement comme sommet initial et terminal.

II - C ne contient pas deux arcs consécutifs qui sont utilisés pour transporter la même demande ou aucune.

III - S'il existe une demande $k \in K$ qui est rechargée en v, alors k est transportée uniquement sur le premier arc entrant de v et sur le dernier arc sortant de v dans C.

IV - Il existe au maximum un rechargement sur v.

V - Le sommet v est traversé au maximum deux fois dans C. De plus, il est traversé deux fois si et seulement s'il y a un rechargement en v.

Preuve. Considérons une solution optimale de cardinalité minimale (C, L). Si C contient deux arcs consécutifs (u, v) et (v, w) sur lesquels une même demande k est transportée, alors on peut construire une solution $(C,', L')$ en remplaçant, dans C et L_k (correspondant au chemin de la demande k), les deux arcs (u, v) et (v, w) par l'arc (u, w). Il est facile de voir que (C', L') est réalisable si (C, L) est réalisable. Comme les coûts vérifient les inégalités triangulaires, alors $c_{(u,w)} \leq c_{(u,v)} + c_{(v,w)}$. Le coût de la solution (C', L') est donc inférieur ou égal à celui de la solution (C, L). De plus, C' contient un arc de moins que C, ce qui implique que (C, L) n'est pas une solution

optimale de cardinalité minimale. Ceci reste vrai si aucune demande n'est transportée sur (u, v) et (v, w). La propriété (II) est donc vérifiée par toute SOCM.

Supposons que dans (C, L), il existe une demande $k \in K$ qui est rechargée en un sommet $v \in V \setminus \{v_0\}$. Supposons également que k n'est pas transportée sur le dernier arc sortant dans C. D'après la propriété (II), le sommet v est donc traversé au minimum trois fois dans C. Une première fois pour décharger la demande k sur v, une deuxième fois pour la recharger et une troisième fois puisque k n'est pas transportée sur le dernier arc sortant de v dans C. Notons par (v_1, v) et (v, v_2) les deux arcs incidents à v sur lesquels la demande k est transportée. Notons par (v, v_3) un arc sortant de v tel que $(v, v_2) \prec_C (v, v_3)$. Considérons le sous-circuit, disons C_1, de C compris entre les arcs (v_1, v) et (v, v_3). C_1 est un circuit commençant en v. Soient D_1 le graphe induit par les arcs de C_1 et L_1 la restriction de L sur D_1. Considérons le triplet (D_1, v, L_1). Le graphe D_1 est clairement Eulérien. De plus, on a $\deg^{\text{out}}(v) \geq 2$ et $(v, v_2) \in \delta^{\text{out}}(v)$. Le triplet (D_1, v, L_1) correspond à une instance du problème du circuit Eulérien avec contraintes de précédence induites par des chemins (voir chapitre 4). De plus, L_1 est un ensemble de chemins arc-disjoints puisque L l'est aussi, ce qui implique que L_1 est sans-Y-sortant et sans-Y-entrant. L'algorithme 6 (page 96) peut donc être utilisé pour résoudre ce problème. De plus, comme C_1 est une solution réalisable de ce problème, alors (D_1, v, L_1) ne contient pas de sous-graphe imprenable. Le premier arc de C_1 est différent de (v, v_2). De plus, il n'existe pas dans L_1 un arc (u, v) de $\delta^{\text{in}}(v)$ avec $(u, v) \prec_C (v, v_2)$. En effet, le seul arc vérifiant cette hypothèse est l'arc (v_1, v) et il n'appartient pas à D^1. Comme les contraintes de précédence sont induites par des chemins, tout arc a vérifiant $a \prec_C (v, v_2)$ vérifie aussi $a \prec_C a'$ ou a' est un arc entrant de v. Cela implique qu'il n'existe pas dans D_1 un arc a tel que $a \prec_C (v, v_2)$. D'après la proposition 4.14, il est possible d'appliquer l'algorithme 6 en choisissant l'arc (v, v_2) comme premier arc du circuit. Notons par C_1' le circuit ainsi obtenu. C_1' contient les mêmes arcs que C_1. De plus, si l'on considère le circuit C' obtenu à partir de C en remplaçant C_1 par C_1', alors (C', L) est une solution réalisable. De plus, C' contient $((v_1, v), (v, v_2))$ comme sous-chemin. D'après la propriété (II), (C', L) n'est pas une SOCM, ce qui implique que (C, L) ne l'est pas non plus. Il s'ensuit que, dans toute solution optimale de cardinalité minimale, si une demande k est rechargée en un sommet v, alors k est transportée sur le dernier arc sortant de v dans C. On peut montrer, d'une manière similaire, que si une demande k est rechargée en un sommet v, alors k est transportée sur le premier arc entrant de v dans C. Il en résulte que la propriété (III) est vérifiée. De même, en utilisant un raisonnement similaire, on peut montrer que s'il existe une demande rechargée en v_0, alors la solution n'est pas de cardinalité minimale.

Supposons qu'il existe un sommet v sur lequel deux demandes sont rechargées.

Comme les chemins des demandes sont arc-disjoints, la propriété (III) n'est pas vérifiée pour au moins une des deux demandes, contredisant ainsi la minimalité de (C, L). La propriété (IV) est donc vérifiée.

Supposons maintenant que C traverse trois fois un sommet. Supposons également qu'il existe une demande k rechargée au sommet v. Comme cette demande est transportée sur le premier arc entrant et le dernier arc sortant de v dans C, il s'ensuit que C contient deux sous-chemins, disons $((u_1, v), (v, v_1))$ et $((u_2, v), (v, v_2))$ sur lequel la demande k n'est pas transportée. De plus, par la propriété (IV), une demande qui n'est pas pas incidente à v n'est pas transportée sur ces sous-chemins. Comme chaque sommet est incident à au plus une demande, il existe donc au plus un arc des deux sous-chemins qui appartient au chemin d'une demande $k \in K$. L'un des deux sous-chemins contient alors deux arcs consécutifs sur lesquels aucune demande n'est transportée, ce qui contredit la propriété (II). Chaque sommet v est donc traversé au plus deux fois par C. De même, si v est traversé deux fois par C et aucune demande n'est rechargée en v, on montre de la même manière que C contient deux arcs consécutifs sur lesquels aucune demande n'est transportée. On déduit alors que tout sommet v est traversé deux fois si et seulement si une demande est rechargée en v. La propriété (V) est donc vérifiée. Finalement, si C traverse un sommet v incident à aucune demande et sur lequel aucune demande n'est rechargée, il est facile de voir que C n'est pas une SOCM. Par hypothèse, v_0 est incident à aucune demande. De plus, aucune demande n'est rechargée en v_0. Il s'ensuit que la propriété (I) est vérifiée. □

Une preuve similaire de la proposition 6.1 est donnée dans le lemme 2.6 et le théorème 2.7 dans [3].

La figure 6.1 donne un exemple de solution réalisable (C, L) vérifiant les propriétés (I)-(V) de la proposition 6.1. Le graphe correspond à l'ensemble des arcs du circuit C. Les arcs en trait plein correspondent aux arcs sur lesquels aucune demande n'est transportée. La légende de la figure indique quel demande est transportée sur les autres arcs. Les numéros sur les arcs indiquent l'ordre dans lequel les arcs sont traversés par le véhicule.

Dans le 1-PRLPA unitaire métrique, chaque arc peut être traversé au maximum une fois par le véhicule. Il pourrait sembler qu'une telle hypothèse restreint tellement l'espace des solutions qu'il peut y avoir des instances pour lesquelles cette hypothèse diminue la qualité des solutions obtenues. On montre maintenant qu'une telle hypothèse n'est pas une restriction. En effet, on montre, dans la proposition suivante, que si l'on considère la variante du 1-PRLPA unitaire métrique dans laquelle le véhicule peut

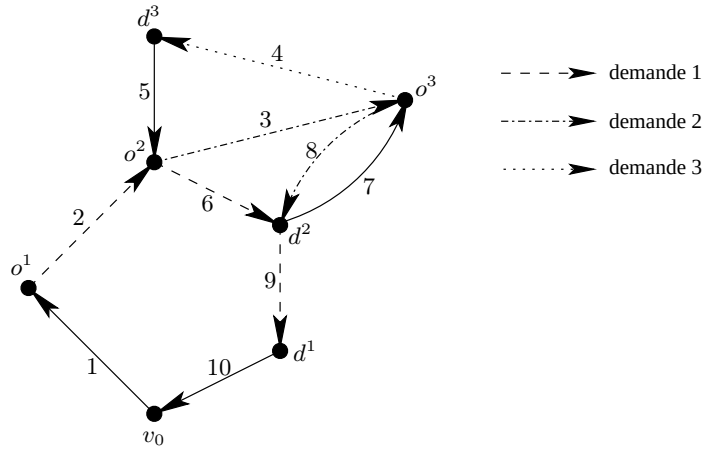

FIGURE 6.1 – Solution vérifiant les propriétés (I)-(V) de la proposition 6.1

traverser plusieurs fois chaque arc, il existe alors toujours une solution optimale dans laquelle chaque arc est traversé au plus une fois par le véhicule.

Proposition 6.2 *Soit (D, v_0, K, c) une instance du 1-PRLPA unitaire métrique. Permettre au véhicule de traverser plusieurs fois un même arc ne change pas la valeur des solutions optimales.*

Preuve. Soit (C, L) une solution optimale de cardinalité minimale pour la variante du 1-PRLPA unitaire métrique où le véhicule peut traverser plusieurs fois chaque arc. Les propriétés (I)-(V) de la proposition 6.1 sont vérifiées par (C, L). Supposons qu'il existe un arc, disons (u, v), qui apparaît deux fois dans C. Par la proposition 6.1.(I), u et v sont différents de v_0. De plus, les propriétés (II) et (V) de la proposition 6.1 impliquent qu'il existe deux demandes distinctes k_u et k_v de K qui sont respectivement rechargées en u et v. Par ailleurs, le sommet u (respectivement v) correspond à l'origine de k_v (respectivement la destination de k_u) à cause de l'unicité des rechargements sur un sommet de $V \setminus \{v_0\}$. L'arc (u, v) apparaît alors dans C la première fois pour transporter k_v. Comme k_v est transportée sur le dernier arc sortant de v par la proposition 6.1.(II), on peut échanger le transport des demandes k_u et k_v sur l'arc (u, v). Cette nouvelle solution est clairement réalisable et l'on a préservé l'optimalité de la solution. Cependant, elle contient deux arcs consécutifs transportant la même demande, ce qui contredit le fait que (C, L) est de cardinalité minimale. □

Cette proposition a une implication intéressante pour le problème de la grue préemptif asymétrique présenté dans l'état de l'art. En effet, la différence entre le 1-PRLPA

unitaire et le PGPA est que dans ce dernier, chaque arc peut être traversé plusieurs fois par le véhicule. Il est alors possible de modifier cette instance - si nécessaire - en une instance du 1-PRLPA unitaire métrique en utilisant les transformations données dans l'introduction de ce chapitre. Dans cette nouvelle instance, résoudre le 1-PRLPA unitaire métrique revient alors à résoudre le PGPA. Tous les résultats de ce chapitre peuvent ainsi s'appliquer au problème de la grue préemptif asymétrique.

On s'intéresse maintenant aux différentes propriétés vérifiées par les solutions optimales de cardinalité minimale en plus de celles de la proposition 6.1. Soit (C, L) une solution optimale de cardinalité minimale. On dit que deux demandes *s'entrecroisent* dans la solution (C, L) si le véhicule transporte la première demande, puis la seconde puis de nouveau la première demande et finalement la seconde. De manière plus formelle, deux demandes distinctes k et k' de K s'entrecroisent s'il existe deux arcs de L_k, disons a_1 et a_2, et deux arcs de $L_{k'}$, disons b_1 et b_2, tels que $a_1 \prec_C b_1 \prec_C a_2 \prec_C b_2$. Comme (C, L) est une solution réalisable, il est clair que l'on a $a_1 \prec_{L_k} a_2$ et $b_1 \prec_{L_{k'}} b_2$. De plus, comme les chemins sont arc-disjoints, les arcs a_1, a_2, b_1 et b_2 sont tous différents. La figure 6.1 montre un exemple de solution (C, L) où des demandes s'intercroisent. Pour le voir, il suffit de considérer les demandes 1 et 2 et les arcs $a_1 = (o^1, o^2)$, $b_1 = (o^2, o^3)$, $a_2 = (o^2, d^2)$ et $b_2 = (o^3, d^2)$.

Dans la proposition suivante, on montre qu'aucune paire de demandes ne s'entrecroisent dans une solution optimale de cardinalité minimale.

Proposition 6.3 *Dans toute solution optimale de cardinalité minimale, il n'existe pas deux demandes qui s'intercroisent.*

Preuve. Soit (C, L) une solution optimale de cardinalité minimale, avec $C = (a_1, a_2, \ldots, a_z)$, $z \geq 1$. Étant donné un entier $i \in \{1, 2, \ldots, z\}$, a_i correspond au $i^{ème}$ arc de C. Supposons que deux demandes s'entrecroisent, disons les demandes 1 et 2. On choisit alors quatre entiers i_1, i'_1, i_2 et i'_2 tels que $a_{i'_k}$ suit directement a_{i_k} dans L_k, $k = 1, 2$, $a_{i_1} \prec_C a_{i_2} \prec_C a_{i'_1} \prec_C a_{i'_2}$, et $i'_2 - i_1$ est minimum. On suppose, sans perte de généralité, que les demandes 1 et 2 forment un *intercroisement minimum*, c'est-à-dire, qu'il n'existe pas de demande k de $K \setminus \{1, 2\}$ et d'entiers i_k et i'_k de $\{1, 2, \ldots, z\}$ tels que a_{i_k} et $a_{i'_k}$ appartiennent à L_k et $a_{i_1} \prec_C a_{i_k} \prec_C a_{i'_1} \prec_C a_{i'_k} \prec_C a_{i'_2}$ ou $a_{i_1} \prec_C a_{i_k} \prec_C a_{i_2} \prec_C a_{i'_k} \prec_C a_{i'_2}$. On pose alors $a_{i_1} = (u_1, u_2)$, $a_{i'_1} = (u_2, u_3)$, $a_{i_2} = (v_1, v_2)$ et $a_{i'_2} = (v_2, v_3)$. Subdivisons C en cinq chemins, notés C_i, $i = 1, 2, \ldots, 5$ de la manière suivante : $C_1 = (a_1, a_2, \ldots, a_{i_1})$, $C_2 = (a_{i_1+1}, a_{i_1+2}, \ldots, a_{i_2})$, $C_3 = (a_{i_2+1}, a_{i_2+2}, \ldots, a_{i'_1-1})$, $C_4 = (a_{i'_1}, a_{i'_1+1}, \ldots, a_{i'_2-1})$ et $C_5 = (a_{i'_2}, a_{i'_2+1}, \ldots, a_z)$. Il est clair que $C = (C_1, C_2, C_3, C_4, C_5)$. Considérons maintenant la séquence $C' = (C_1, C_4, C_3, C_2, C_5)$. Il est facile de voir que

C' correspond également à un circuit commençant en v_0. Supposons qu'il existe une demande, disons la demande 3, et deux arcs a_{i_3} et $a_{i'_3}$ de L_3 tels que C' ne respecte pas la contrainte de précédence induite par ces deux arcs, c'est-à-dire, $a_{i_3} \prec_{L_3} a_{i'_3}$ et $a_{i'_3} \prec_C a_{i_3}$. Dans ce cas, comme C est réalisable, on a alors soit $a_{i_3} \in C_2$ et $a_{i'_3} \in C_3$, soit $a_{i_3} \in C_2$ et $a_{i'_3} \in C_4$, soit $a_{i_3} \in C_3$ et $a_{i'_3} \in C_4$. Dans les deux premiers cas (respectivement le dernier cas), on obtient que les demandes 2 et 3 (respectivement 1 et 3) s'intercroisent. De plus, on a $a_{i_1} \prec_C a_{i_3} \prec_C a_{i'_2}$ (respectivement $a_{i_1} \prec_C a_{i'_3} \prec_C a_{i'_2}$), ce qui contredit l'hypothèse selon laquelle les demandes 1 et 2 forment un intercroisement minimum. La solution (C', L) est donc une solution réalisable du 1-PRLPA unitaire métrique de même coût que (C, L), ce qui implique que (C', L) est une solution optimale. Cependant, comme a_{i_1}, $a_{i'_1}$ et a_{i_2}, $a_{i'_2}$ sont respectivement consécutifs dans C', par la propriété 6.1.(II), (C', L) n'est pas de cardinalité minimale. Par conséquent, (C, L) n'est pas une solution optimale de cardinalité minimale. □

La figure 6.2 montre le nouveau circuit obtenu à partir de celui de la figure 6.1 lorsque l'on applique la transformation donnée dans la preuve de la proposition 6.3. La transformation du cicuit de la figure 6.1 considère les demandes les demandes 1 et 2 et les quatre entiers $i_1 = 2$, $i_2 = 3$, $i'_1 = 6$ et $i'_2 = 8$. Ce nouveau circuit C ne contient plus de demandes s'intercroisant. Par contre, la solution (C, L) n'est pas de cardinalité minimale puisqu'une demande est transportée sur le sous-chemin $((o^1, o^2), (o^2, d^2))$ et une autre sur le sous-chemin $((o^2, o^3), (o^3, d^2))$. La figure 6.3 donne la nouvelle solution obtenue en remplaçant respectivement les sous-chemins $((o^1, o^2), (o^2, d^2))$ et $((o^2, o^3), (o^3, d^2))$ par les arcs (o^1, d^2) et (o^2, d^2).

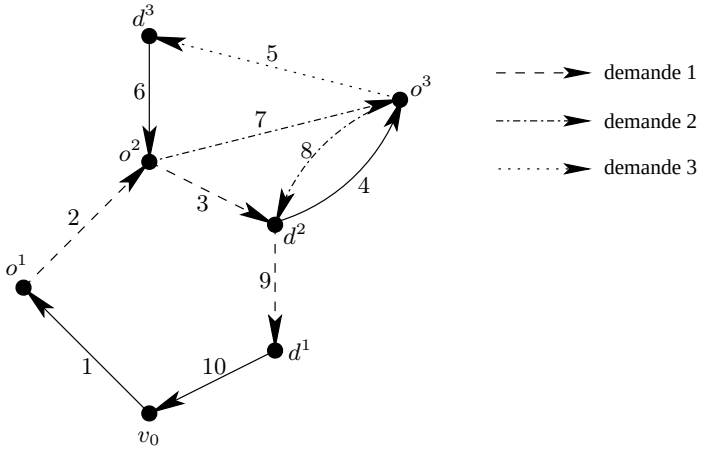

FIGURE 6.2 – Solution obtenue à partir de celle donnée par la figure 6.1 par application de la transformation donnée dans la preuve de la preuve 6.3

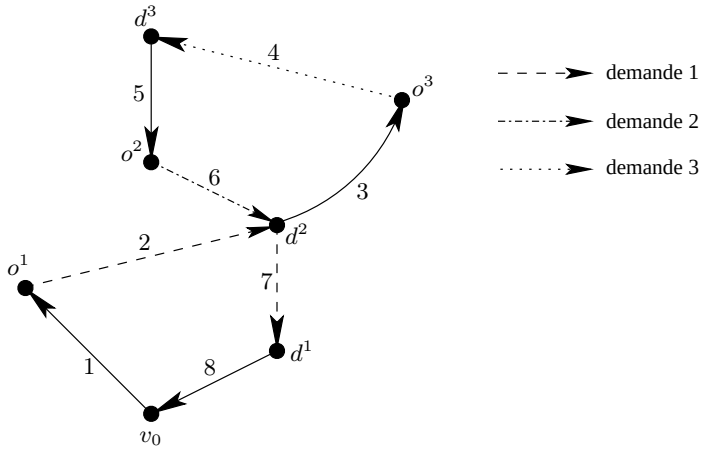

FIGURE 6.3 – Solution obtenue à partir de celle donnée par la figure 6.2 en supprimant les arcs consécutifs utilisés pour le transport d'une même demande

On donne maintenant quelques définitions et notations. Un *cactus orienté* est un graphe orienté connexe dans lequel chaque arc appartient à exactement un circuit élémentaire. Soit (C, L) une solution optimale de cardinalité minimale et v un sommet traversé plusieurs fois par C. On note par C_{vv} le sous-circuit de C commençant en v. Comme (C, L) est une SOCM, alors, d'après la propriété 6.1.(V), C_{vv} est défini de manière unique pour tout sommet v traversé plusieurs fois par C. On étend la définition d'intercroisement donnée pour les demandes dans le cas des sommets. Soient u et v deux sommets traversés par un circuit C. On dit que u et v *s'intercroisent* si C traverse une première fois u puis v puis une deuxième fois u puis v.

Proposition 6.4 *Soit (D, v_0, K, c) une instance du 1-PRLPA unitaire métrique. Toute solution optimale de cardinalité minimale (C, L) de (D, v_0, K, c) est telle que le graphe induit par les arcs de C est un cactus orienté.*

Preuve. Supposons que le graphe $D_C = (V_C, A_C)$ induit par les arcs de C n'est pas un cactus. Comme D_C est Eulérien, il existe alors un arc (u_1, v_1) de A_C et deux circuits élémentaires différents, disons C_1 et C_2, tels que $(u_1, v_1) \in C_1 \cap C_2$. Supposons, sans perte de généralité, que les circuits C_1 et C_2 commencent par l'arc (u_1, v_1), *i.e.*,

$$C_1 = ((u_1, v_1), s_1, s_2, \ldots, s_{z_1}) \text{ et } C_2 = ((u_1, v_1), t_1, t_2, \ldots, t_{z_2}).$$

Soit $i_1 \in \{1, 2, \ldots, \min\{z_1, z_2\}\}$ l'entier tel que pour tout $i = 1, 2, \ldots, i_1 - 1$, on a $s_i = t_i$ et $s_{i_1} \neq t_{i_1}$. Notons par u_2 l'extrémité initiale de l'arc s_{i_1}. Soit v_2 le premier sommet différent de u_2, apparaissant à la fois dans $C_1 \setminus \{((u_1, v_1), s_1, s_2, \ldots, s_{i_1-1})\}$

et $C_2 \setminus \{((u_1, v_1), t_1, t_2, \ldots, t_{i_1-1})\}$. Comme C_1 et C_2 sont différents, alors i_1 existe. Le sommet v_2 existe également puisque les deux circuits se terminent en u_1. Par construction, les sous-chemins de u_2 à v_2 dans C_1 et C_2 correspondent à deux chemins sommet-disjoints de u_2 à v_2. On note $P_1 = ((u_2, \alpha_1), (\alpha_1, \alpha_2) \ldots, (\alpha_{l_1}, v_2))$ et $P_2 = ((u_2, \beta_1), (\beta_1, \beta_2) \ldots, (\beta_{l_2}, v_2))$ ces deux chemins. On suppose, sans perte de généralité, que l'arc $(u_2, \alpha_1) \prec_C (u_2, \beta_1)$.

On montre maintenant qu'il existe dans C deux sommets, disons x_1 et x_2, s'intercroisant dans C. Supposons le contraire. Dans ce cas, comme u_2 et v_2 ne s'intercroisent pas, ceci implique que le sommet u_2 n'appartient pas à $C_{v_2v_2}$ ou que le sommet v_2 n'appartient pas à $C_{u_2u_2}$. Considérons le sous-circuit $C_{u_2u_2}$ de C. D'après la propriété 6.1.(V), $\delta^{\text{out}}(u_2) = \{(u_2, \alpha_1), (u_2, \beta_1)\}$. Comme $(u_2, \alpha_1) \prec_C (u_2, \beta_1)$, le premier arc de $C_{u_2u_2}$ est (u_2, α_1). Supposons que (α_1, α_2) n'appartient pas à $C_{u_2u_2}$. Comme $C_{u_2u_2}$ est un circuit et qu'il contient l'arc (u_2, α_1), $C_{u_2u_2}$ traverse α_1. C traverse alors u_2, α_1, puis u_2. Comme (α_1, α_2) n'appartient pas à $C_{u_2u_2}$ mais appartient à C, C traverse α_1 avant ou après $C_{u_2u_2}$. Par conséquent, les sommets α_1 et u_2 s'intercroisent, une contradiction. (α_1, α_2) est donc un arc de $C_{u_2u_2}$. De la même manière, on montre que si l'arc (α_{i-1}, α_i) appartient au chemin $C_{u_2u_2}$, alors l'arc (α_i, α_{i+1}) appartient également à $C_{u_2u_2}$. Dans le cas contraire, les sommets u_2 et α_i s'intercroisent. On déduit alors par récurrence que v_2 appartient au chemin de u_2 à u_2 dans C. De la même manière, on montre que u_2 appartient au chemin $C_{v_2v_2}$, ce qui implique que u_2 et v_2 s'intercroisent. Il existe donc deux sommets x_1 et x_2 qui s'intercroisent dans C.

Comme (C, L) est une solution de cardinalité minimale et x_1 et x_2 apparaissent deux fois dans C, alors x_1 et x_2 sont des sommets de rechargements et il existe deux demandes différentes, disons k et k', qui sont respectivement rechargées en x_1 et x_2 (propriété (V)). De plus, la demande k (respectivement k') est transportée sur le premier arc entrant et le premier arc sortant de x_1 (respectivement x_2) dans C (propriété 6.1.(III)). On déduit alors que les demandes k et k' s'intercroisent. D'après la proposition 6.3, (C, L) n'est pas une solution de cardinalité minimale. □

Dans le reste de ce chapitre, étant donné un cactus orienté $D_C = (V_C, A_C)$, induit par les arcs du circuit C d'une SOCM (C, L), on note par $\alpha + 1$ le nombre de circuits élémentaires consitituant le cactus orienté. Ces circuits sont notés $C_0, C_1, \ldots, C_\alpha$. C_0 est le circuit élémentaire passant par v_0. De plus, on associe à chaque circuit C_i un sommet d'origine v_i pour tout $i = 0, 1, \ldots, \alpha$ de la manière suivante. Le sommet v_0 est le sommet d'origine associé au circuit C_0. Le sommet d'origine v_i associé au circuit C_i, $i = 1, 2, \ldots, \alpha$, est le sommet v de C_i tel que sa suppression dans D_C induit deux composantes connexes, l'une contenant les arcs de $C_i \setminus \{v\}$, l'autre contenant le sommet

v_0. Chaque sommet d'origine différent de v_0 correspond à un sommet d'articulation du circuit. Il est facile de voir qu'il existe pour tout circuit C_i, $i = 0, 1, \ldots, \alpha$, un unique sommet d'origine. De plus, à chaque sommet d'origine v_i est associé un unique circuit élémentaire car tout sommet v de $V \setminus \{v_0\}$ apparaît au plus deux fois dans C et v_0 apparaît dans C uniquement comme sommet initial et terminal. Par ailleurs, étant donné un cactus orienté D_C induit par une SOCM (C, L), il existe un unique ordre sur les arcs de D_C correspondant à un circuit C commençant en v_0. De plus, pour tout circuit élémentaire C_i, l'ordre dans lequel ses arcs sont traversés à partir du sommet d'origine v_i par C correspond à l'ordre induit par le parcours de C_i à partir du sommet d'origine C_i.

On montre, dans le théorème suivant, que résoudre le 1-PRLPA unitaire métrique consiste à déterminer un sous-graphe vérifiant certaines propriétés.

Théorème 6.5 *Le 1-PRLPA unitaire métrique consiste, étant donnée une instance (D, v_0, K, c), à déterminer un sous-graphe orienté $D_C = (V_C, A_C)$ de coût minimum tel que*

1 - D_C est un cactus orienté (sans perte de généralité, on suppose que D_C est composé de $\alpha + 1$ circuits élémentaires $C_0, C_1, \ldots, C_\alpha$ ayant respectivement pour sommet d'origine $v_0, v_1, \ldots, v_\alpha$),

2 - $\deg^{\text{out}}_{D_C}(v_0) = 1$ et $\deg^{\text{out}}_{D_C}(v) \leq 2$ pour tout $v \in V \setminus \{v_0\}$,

3 - pour toute demande k de K, il existe un circuit C_i, $i \in \{0, 1, \ldots, \alpha\}$ contenant les sommets o^k et d^k,

4 - pour toute demande k de K, le sommet d'origine v_i, du circuit élémentaire C_i contenant les sommets o^k et d^k, se trouve sur le chemin élémentaire de d^k à o^k dans C_i (il peut être identifié à o^k ou d^k),

5 - Pour toute paire de demandes $k \neq k'$ de K, il existe deux chemins élémentaires arc-disjoints, l'un de o^k à d^k, l'autre de $o^{k'}$ à $d^{k'}$.

Preuve. On montre tout d'abord que toute solution optimale de cardinalité minimale peut être associée à un sous-graphe D_C vérifiant les différentes propriétés du théorème. Soit (C, L) une SOCM. Soit $D_C = (V_C, A_C)$ le graphe induit par les arcs de C. La proposition 6.4 assure que D_C est un cactus orienté. Les propriétés (I) et (V) de la proposition 6.1 impliquent que $\deg^{\text{out}}_{D_C}(v_0) = 1$ et $\deg^{\text{out}}_{D_C}(v) \leq 2$ pour tout $v \in V \setminus \{v_0\}$. Supposons maintenant qu'il existe une demande k dont l'origine et la destination ne se trouvent pas sur un même circuit élémentaire. Il existe alors deux circuits élémentaires connexes C_i et C_j, $i \neq j$, tels que $C_i \cap L_k \neq \emptyset$ et $C_j \cap L_k \neq \emptyset$. ($L_k$ est le chemin de

la demande k de o^k à d^k). Le sommet v incident aux deux circuits élémentaires est le sommet d'origine de C_i ou de C_j. Comme D_C est un cactus orienté, alors C traverse le sommet v uniquement lors du passage de C_i à C_j et du passage de C_j à C_i. Le chemin L_k contient alors deux arcs consécutifs dans C, ce qui contredit l'hypothèse 6.1.(II). On déduit alors que le chemin L_k d'une demande $k \in K$ correspond au chemin de o^k à d^k dans le circuit élémentaire de D_C contenant ces deux sommets puisque L_k est un chemin élémentaire et D_C est un cactus orienté. La propriété (3) est donc vérifiée. Supposons maintenant que la propriété (4) n'est pas vérifiée. Autrement dit, pour une demande k de K, le circuit C_i passant par o^k et d^k est tel que son sommet d'origine v_i n'appartient pas au chemin de d^k à o^k dans C_i. On déduit alors que v_i est un sommet du chemin de o^k à d^k dans le circuit C_i et v_i est différent de o^k et d^k. Comme on l'a déjà mentionné, le chemin L_k de la demande k est le chemin de o^k à d^k dans C_i. De plus, l'ordre des arcs de C_i dans C est équivalent à celui obtenu en parcourant C_i à partir de v_i. On obtient donc que le premier arc de L_k traversé par C est l'arc ayant v_i comme extrémité initiale. Ceci implique que cet arc est traversé avant le premier arc de L_k. Le circuit C ne respecte alors pas les contraintes de précédence induites par L_k. (C, L) n'est donc pas une solution réalisable, ce qui contredit l'hypothèse. La dernière propriété est vérifiée par (C, L) car L est un ensemble de chemins élémentaires arc-disjoints.

Montrons maintenant que tout sous-graphe D_C vérifiant les propriétés (1)-(5) peut être associé à une solution réalisable du 1-PRLPA unitaire métrique. Comme on l'a déjà mentionné, il existe un unique ordre sur les arcs de D_C formant un circuit Eulérien commençant en v_0 et un unique chemin élémentaire de l'origine à la destination de toute demande de K. On peut donc associer à tout sous-graphe D_C une unique solution (C, L). Il faut maintenant s'assurer que (C, L) est réalisable. Il est clair que (C, L) est réalisable si et seulement si C respecte les contraintes de précédence induites par L et L est un ensemble de chemins arc-disjoints. La propriété (4) implique qu'il n'existe pas de demande k de K dont le chemin élémentaire L_k passe par le sommet d'origine v_i associé au circuit élémentaire C_i de D_C contenant o^k et d^k. L'ordre dans lequel les arcs de L_k sont traversés par C correspond alors à celui de L_k, ce qui implique que C respecte les contraintes de précédences induites par L. Finalement, comme D_C est un cactus orienté, il existe, pour toute demande $k \in K$, un unique chemin de o^k à d^k. La propriété (5) implique alors que les chemins sont arc-disjoints.

Finalement, comme le coût d'une solution (C, L) correspond à la somme des coûts des arcs du graphe D_C, trouver une SOCM de coût minimum correspond à trouver un sous-graphe D_C de coût minimum vérifiant les propriétés (1)-(5). □

Le théorème 6.5 a des conséquences importantes au niveau de l'information nécessaire

pour décrire une solution du 1-PRLPA unitaire métrique. En effet, dans le chapitre 4, on a montré que toute solution du 1-PRLPA unitaire devait être décrite par l'ensemble des arcs traversés par le véhicule et les chemins des demandes. Cependant, le 1-PRLPA unitaire métrique consiste à déterminer un sous-graphe vérifiant certaines propriétés, les arcs du sous-graphe correspondant aux arcs traversés par le véhicule. On a également montré qu'étant donné ce sous-graphe, il n'existe qu'un chemin élémentaire possible pour chaque demande. L'information relative aux arcs traversés par les demandes peut donc être facilement déterminée à partir des arcs traversés par le véhicule. Cette information n'est donc plus nécessaire dans la description des solutions. On présente, dans la section suivante, un programme linéaire en nombres entiers permettant de formuler le 1-PRLPA unitaire métrique.

6.2 Programme linéaire en nombres entiers

Dans cette partie, on donne un programme linéaire en nombres entiers pour résoudre le 1-PRLPA unitaire métrique. Comme on l'a déjà mentionné dans le début de ce chapitre, le problème d'optimisation consiste à déterminer un sous-graphe orienté de D, induit par les arcs traversés par le véhicule, vérifiant les propriétés 1-5 du théorème 6.5. On a également montré que l'information relative aux arcs traversés par les demandes n'est plus nécessaire pour décrire les solutions du problème. On introduit donc les variables suivantes

$$y_a = \begin{cases} 1 & \text{si le véhicule traverse l'arc } a, \\ 0 & \text{sinon,} \end{cases}$$

pour tout arc $a \in A$. Comme D est complet, alors $y \in \{0,1\}^{n(n-1)}$ où n est le nombre de sommets de D. Contrairement aux formulations données dans le chapitre 4, on ne considère plus les variables x associées aux chemins des demandes. Soit $S_{UM}(D, v_0, K, c)$ l'ensemble des vecteurs y associés aux solutions réalisables du 1-PRLPA unitaire mé-

trique. Considérons les contraintes suivantes :

$$\sum_{a\in\delta^{\text{out}}(W)} y_a - y_{a'} \geq 0 \qquad \forall\ W \subseteq V \text{ tel que } v_0 \in W, \forall\ a' \in A(\overline{W}), \tag{6.1}$$

$$\sum_{a\in\delta^{\text{out}}(v)} y_a - \sum_{a\in\delta^{\text{in}}(v)} y_a = 0 \qquad \forall\ v \in V, \tag{6.2}$$

$$y(P_1) + y(P_2) \leq |P_1| + |P_2| - 1 \qquad \forall u \neq v \in V, \forall P_1, P_2 \in \mathcal{P}_{uv} \text{ sommet-disjoints} \tag{6.3}$$

$$y(\delta^{\text{out}}(v_0)) = 1, \tag{6.4}$$

$$y(\delta^{\text{out}}(v)) \leq 2 \qquad \forall\ v \in V \setminus \{v_0\}, \tag{6.5}$$

$$y(\delta_{D\setminus v}(W)) \geq 1 \qquad \begin{array}{l}\forall k \in K, \forall v \in V \setminus \{o^k, d^k\}, \forall W \subseteq V \setminus \{v\},\\ o^k \in W, d^k \notin W,\end{array} \tag{6.6}$$

$$y(\delta_{D\setminus\{\delta^{\text{out}}(o^k)\cup\delta^{\text{in}}(d^k)\}}(W)) \geq 1 \qquad \forall k \in K, \forall W \subseteq V, v_0 \in W, o^k, d^k \notin W, \tag{6.7}$$

$$y(\delta^{\text{out}}_{D\setminus v_0}(W)) \geq 2 \qquad \begin{array}{l}\forall k \neq k' \in K, \forall W \subseteq V, o^k, o^{k'} \in W,\\ d^k, d^{k'} \notin W,\end{array} \tag{6.8}$$

$$y_a \geq 0 \qquad \forall\ a \in A, \tag{6.9}$$

$$y_a \leq 1 \qquad \forall\ a \in A, \tag{6.10}$$

où $\mathcal{P}_{uv}$ est l'ensemble des chemins élémentaires de u à v, u et v étant deux sommets distincts du graphe. (Deux uv-chemins élémentaires sont sommet-disjoints si chaque sommet différent de u et v, appartient à au plus un des deux chemins.) On montre dans le théorème suivant que ces contraintes sont nécessaires et suffisantes pour définir les solutions entières du 1-PRLPA unitaire métrique.

Théorème 6.6 *L'ensemble* $\{y \in \{0,1\}^{n(n-1)} : y \text{ satisfait } (6.1) - (6.8)\}$ *correspond à l'ensemble* $S_{UM}(D, v_0, K, c)$ *des solutions réalisables du 1-PRLPA unitaire métrique.*

Preuve. On montre tout d'abord que tout vecteur binaire de $S_{UM}(D, v_0, K, c)$ vérifie les contraintes (6.1)-(6.8). Soit $y \in \{0,1\}^{n(n-1)}$ un vecteur de $S_{UM}(D, v_0, K, c)$. Notons par D_C le graphe induit par les arcs a de A tels que $y_a = 1$. D'après le théorème 6.5, D_C est un cactus orienté, ce qui implique que D_C est Eulérien. Les contraintes (6.1) et (6.2) sont donc vérifiées par y. Considérons les contraintes (6.3). Pour montrer que ces contraintes sont vérifiées par y, il faut prouver que si D_C est un cactus, alors il ne contient pas deux sommets distincts u et v de V et deux chemins élémentaires sommet-disjoints de u à v. Supposons que ceci ne soit pas vrai, c'est-à-dire, il existe dans D_C deux chemins sommet-disjoints, disons P_1 et P_2, de u à v. Comme D_C est Eulérien, dans $D_C \setminus \{P_1 \cup P_2\}$, le nombre d'arcs entrants et le nombre d'arcs sortants de chaque sommet sont égaux, excepté pour les sommets u et v. De manière plus précise,

on a $\deg^{\text{out}}(u) - \deg^{\text{in}}(u) = -2$ et $\deg^{\text{out}}(v) - \deg^{\text{in}}(v) = 2$. Il existe donc un chemin de v à u dans $D_C \setminus \{P_1 \cup P_2\}$, disons P_3. Les circuits $P_1 \cup P_3$ et $P_2 \cup P_3$ sont différents et contiennent au moins un arc en commun, ce qui implique que D_C n'est pas un cactus orienté. Les contraintes (6.3) sont donc vérifiées par y. Comme D_C vérifie la propriété 6.5.(2), y satisfait les contraintes (6.4) et (6.5). La propriété 6.5.(3) assure que l'origine et la destination de chaque demande k de K appartiennent au même circuit élémentaire. Il existe donc un chemin élémentaire de o^k à d^k et un autre de d^k à o^k qui sont sommet-disjoints. Si l'on supprime un sommet différent de o^k et d^k dans ce circuit, il est clair qu'il reste un des deux chemins, ce qui implique, par le théorème de Menger [65], que toute coupe séparant o^k et d^k dans D_C est non vide. Les contraintes (6.6) sont donc satisfaites par y. Considérons les contraintes (6.7). Supposons que y ne vérifie pas ces contraintes, c'est-à-dire, il existe une demande k' de l'ensemble K et un sous-ensemble de sommets $W' \subseteq V$ contenant v_0 mais pas $o^{k'}$ ni $d^{k'}$ tels que

$$y(\delta_{D \setminus \{\delta^{\text{out}}(o^{k'}) \cup \delta^{\text{in}}(d^{k'})\}}(W')) = 0. \tag{6.11}$$

Comme D_C est Eulérien et contient les sommets v_0, $o^{k'}$ et $d^{k'}$, on a $\deg^{\text{out}}_{D_C}(W) \geq 1$ et $\deg^{\text{in}}_{D_C}(W) \geq 1$ pour tout sous-ensemble W de sommets contenant v_0 mais pas $o^{k'}$ ni $d^{k'}$. On déduit alors que W' est tel que $\delta_{D_C}(W')$ est inclus dans $\delta^{\text{out}}_{D_C}(o^{k'}) \cup \delta^{\text{in}}_{D_C}(d^{k'})$. Notons par C_i le circuit élémentaire contenant $o^{k'}$ et $d^{k'}$. La propriété 6.5.(3) assure l'existence d'un tel circuit. De plus, comme D_C est un cactus orienté, alors ce circuit est unique. D'après l'équation (6.11), on déduit que l'intersection entre W' et C_i correspond exactement à l'ensemble des sommets intermédiaires du chemin élémentaire de $o^{k'}$ à $d^{k'}$ dans C_i. Supposons que le sommet d'origine v_i de C_i n'appartient pas à W'. Dans ce cas, la composante connexe dans $D \setminus v_i$ contenant v_0 contient aussi des arcs de $C_i \setminus v_i$ puisqu'elle contient W' et $\delta_{D_C}(W')$ est inclus dans $\delta^{\text{out}}_{D_C}(o^{k'}) \cup \delta^{\text{in}}_{D_C}(d^{k'})$. Ceci contredit alors la définition de v_i. Le sommet d'origine v_i appartient donc à W'. Ceci implique que v_i appartient au chemin de $o^{k'}$ à $d^{k'}$ et que v_i est différent de $o^{k'}$ et $d^{k'}$, ce qui contredit la propriété 6.5.(4). Finalement, d'après la propriété 6.5.(5), pour toute paire de demandes distinctes k et k', il existe dans D_C deux chemins élémentaires arc-disjoints, l'un de o^k à d^k et l'autre de $o^{k'}$ à $d^{k'}$. Par ailleurs, comme D_C est un cactus, il existe un unique chemin élémentaire de l'origine à la destination d'une demande. De plus, v_0 n'appartient pas à ce chemin. En effet, si l'origine o^k et la destination d^k d'une demande k appartiennent à un circuit élémentaire différent de C_0, alors comme v_0 n'appartient pas à ce circuit élémentaire et le chemin L_k utilise uniquement des arcs de ce circuit, alors v_0 n'appartient pas à L_k. Si o^k et d^k appartiennent à C_0, la propriété 6.5.(4) assure que L_k ne passe pas par v_0. Par conséquent, les contraintes (6.8) sont vérifiées par y.

Considérons maintenant un vecteur $y \in \{0,1\}^{n(n-1)}$ vérifiant les contraintes (6.1)-(6.8). On montre maintenant que le graphe induit par les arcs de D avec $y_a = 1$, noté

par la suite D_y, est un cactus orienté vérifiant les propriétés du théorème 6.5. Il est clair que le graphe D_y est Eulérien. De plus, d'après les contraintes (6.3), il n'existe pas deux sommets distincts u et v de V avec deux chemins élémentaires sommet-disjoints de u à v, ce qui implique que D_y est un cactus orienté. (La preuve impliquant qu'un graphe ne correspondant pas à un cactus orienté possède deux chemins élémentaires sommet-disjoints de u à v, où u et v sont deux sommets distincts du graphe, est donnée dans la preuve de la proposition 6.4.) Les contraintes (6.4) et (6.5) impliquent que la propriété 6.5.(2) est vérifiée. Supposons maintenant que le graphe D_y ne vérifie pas la propriété 6.5.(3). Cela signifie qu'il existe une demande k dont l'origine et la destination n'appartiennent pas à un même circuit élémentaire. Comme o^k et d^k appartiennent à deux circuits différents et D_y est un cactus orienté, il s'ensuit qu'il existe un sommet déconnectant le graphe en deux composantes connexes, l'une contenant o^k, l'autre contenant d^k. Ceci implique qu'une contrainte de type (6.6) est violée par y. Supposons maintenant que le graphe D_y ne vérifie pas la propriété 6.5.(4), c'est-à-dire, il existe une demande $k \in K$ tel que le sommet d'origine v_i du circuit élémentaire C_i contenant o^k et d^k est différent de o^k et d^k et appartient au chemin de o^k à d^k dans C_i. Notons par a_1 l'arc sortant de o^k dans C_i et par a_2 l'arc entrant de d^k dans C_i. La suppression de a_1 et a_2 dans D_y induit deux composantes connexes. Dans le cas contraire, comme D_y est Eulérien, cela implique que $D_y \setminus \{a_1, a_2\}$ est fortement connexe, et il existe alors dans $D_y \setminus \{a_1, a_2\}$ un chemin de l'extrémité initiale de a_1 à son extrémité terminale, ce qui implique que les contraintes (6.3) ne sont pas respectées. Notons par W^* la composante connexe contenant v_0. Alors on a $\delta_{D_y}(W^*) = \{a_1 \cup a_2\}$. La contrainte (6.7) associée à W^* n'est donc pas satisfaite par y. La propriété 6.5.(4) est donc vérifiée par D_y. Finalement, le théorème de Menger implique que D_y satisfait la dernière proposition grâce aux contraintes (6.8), ce qui termine la preuve. □

Le théorème précédent permet de formuler le 1-PRLPA unitaire métrique à l'aide du programme linéaire en nombres entiers suivant

$$\min\{c^T y \mid y \in \mathrm{conv}(S_{UM}(D, v_0, K, c))\},$$

noté par la suite P_{UM}.

Théorème 6.7 *La relaxation linéaire de P_{UM} peut être résolue en temps polynomial.*

Preuve. Tout d'abord, comme les contraintes (6.2), (6.4), (6.5), (6.9) et (6.10) sont en nombre polynomial dans P_{UM}, la complexité de la relaxation linéaire dépend uniquement de la complexité du problème de séparation des inégalités (6.1), (6.3), (6.6)-(6.8).

Notons par $\bar{y}$ la solution à séparer.
La séparation des contraintes (6.1) peut se ramener au calcul d'un nombre polynomial de coupes minimum. Pour tout arc $a \in A$ n'appartenant pas à $\delta(v_0)$, la séparation des contraintes (6.1) associées à a se ramène à déterminer une v_0v^a-coupe minimum où v^a est le sommet obtenu par contraction de a, la fonction poids étant donnée par $\bar{y} \geq 0$.
Considérons les contraintes (6.3) associées à deux sommets u et v. Considérons le graphe D' obtenu à partir de D de la manière suivante : chaque sommet $w \in V \setminus \{u, v\}$ est remplacé par deux nouveaux sommets w_s et w_t ; chaque arc $e = (s, w)$ (respectivement $e = (w, t)$) est remplacé par un arc (s, w_s) (respectivement (w_t, t)) de coût $1 - \bar{y}_e$ et de capacité 1 ; l'arc (w_s, w_t) de coût 0 et de capacité 1 est ajouté. Dans D', on calcule un uv-flot de valeur 2 de coût minimum. Un tel flot se décompose en deux chemins arc-disjoints dans D' correspondant à deux chemins sommet-disjoints P_1, P_2 dans D. Le coût du flot vaut $|P_1| + |P_2| - y(P_1) - y(P_2)$. Ainsi, il est clair qu'une contrainte (6.3) associée à u et v est violée par $\bar{y}$ si et seulement si le coût du flot est strictement inférieur à 1. La problème de séparation des contraintes (6.3) se ramène alors à $\frac{n(n-1)}{2}$ calcul de flot de coût minimum.
Le problème de séparation des contraintes (6.3) associées à deux sommets distincts u et v se ramène à déterminer un flot de volume 2 de coût minimum dans un graphe orienté auxiliaire obtenu en remplaçant chaque sommet par un arc de capacité un et de coût nul, la capacité des arcs $a \in A$ étant fixée à un et le coût étant égal à 1 - y_a. Le problème de séparation des contraintes (6.3) se ramène alors à calculer un flot de coût minimum pour tout couple de sommets distincts du graphe.
Considérons les contraintes (6.6) associées à une demande $k \in K$ et à un sommet $v \in V \setminus \{o^k, d^k\}$. La séparation de ces contraintes consiste à déterminer une coupe minimum dans $D \setminus v$ dont les poids sont donnés par le vecteur $\bar{y} \geq 0$. La séparation des contraintes (6.6) se ramène alors à au plus $(n-2)p$ calculs de coupe minimum.
Les contraintes (6.7) associées à une demande $k \in K$ consiste à déterminer une v_0o^k-coupe minimum dans le graphe D avec des poids infinis pour les arcs (o^k, d^k) et (d^k, o^k), les poids restants étant donnés par $\bar{y} \geq 0$. Le problème de séparation des contraintes (6.7) se ramène au calcul de p coupes minimum.
La séparation des contraintes (6.8) associées à deux demandes $k \neq k'$, consiste à déterminer s'il existe dans $D \setminus v_0$ un ensemble $W \subset V$ contenant les origines des deux demandes mais pas les destinations de ces demandes, tel que la coupe sortante de W soit strictement inférieur à 2. Le problème de séparation de ces contraintes se ramène au calcul de $\frac{p(p-1)}{2}$ coupes minimum.
La séparation des contraintes (6.1), (6.3), (6.6)-(6.8) se ramène à un nombre polynomial de calcul de coupes minimum et de flot de coût minimum dont les poids sont positifs. Comme ces deux problèmes sont polynomiaux, le problème de séparation associé aux contraintes (6.1), (6.3), (6.6)-(6.8) peut être résolu en temps polynomial. La relaxation

linéaire de P_{UM} peut donc être résolue en temps polynomial. □

6.3 Résolution heuristique

Dans cette section, on propose une métaheuristique pour le 1-PRLPA unitaire métrique. On présente d'abord comment les solutions sont codées dans cet algorithme. On présente ensuite une heuristique permettant d'obtenir une solution initiale. La métaheuristique, permettant d'améliorer cette solution initiale, est ensuite décrite. On discute finalement des résultats obtenus.

6.3.1 Codage des solutions

On vient de voir, dans la section précédente, que seule l'information relative aux arcs traversés par le véhicule est nécessaire pour décrire les solutions du 1-PRLPA unitaire métrique. Par conséquent, seule cette information sera utilisée pour le codage des solutions dans l'heuristique. La question réside maintenant dans le codage efficace, d'un point de vue algorithmique, des arcs traversés par le véhicule.

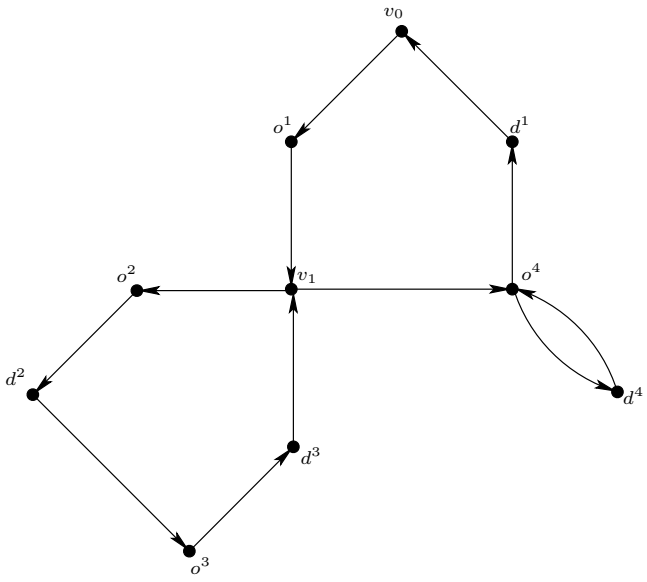

FIGURE 6.4 – Exemple d'une solution optimale de cardinalité minimale

Il est possible de coder toute SOCM à l'aide de la séquence des sommets traversés par le véhicule. Cependant, on peut tirer parti du fait que le graphe induit par les

arcs du véhicule vérifie certaines propriétés. En effet, puisque le graphe D_C, induit par les arcs du trajet du véhicule, est un cactus orienté, on peut coder la séquence des sommets de chaque circuit élémentaire obtenue en parcourant le circuit à partir du sommet d'origine. Ces différentes séquences sont reliées entre elles par les sommets d'origine. On obtient alors une v_0-arborescence. Étant donnée la solution optimale de cardinalité minimale donnée par la figure 6.4, on obtient alors la figure 6.5.

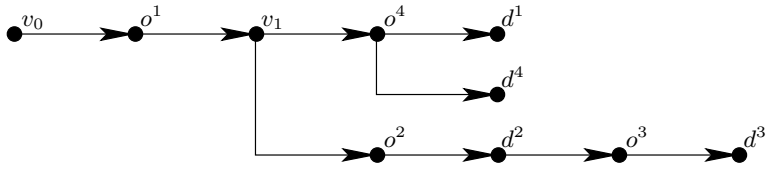

FIGURE 6.5 – Codage des séquences de sommets de chaque circuit élémentaire de la SOCM donnée par la figure 6.4

Si cette structure de données permet de représenter toutes les solutions optimales du 1-PRLPA unitaire métrique, il est cependant possible d'améliorer le codage de ces solutions en tenant compte des autres propriétés satisfaites par le graphe D_C. En effet, on sait que le chemin L_k de chaque demande $k \in K$ correspond au chemin élémentaire de o^k à d^k dans le circuit élémentaire contenant ces deux sommets. On peut alors, dans le codage, différencier dans les circuits les arcs appartenant à un chemin d'une demande des autres arcs, c'est-à-dire, coder les chemins des demandes et les circuits élémentaires séparément. On code chaque chemin d'une demande k à l'aide de la séquence des sommets de L_k différents de l'origine o^k et de la destination d^k. Cette séquence est rattachée à l'origine de la demande. Si une origine ou une destination est également sommet de rechargement (c'est-à-dire, un sommet d'origine d'un circuit élémentaire), on crée une copie fictive de ce sommet afin que chaque circuit contienne l'origine et la destination des demandes transportées sur ce circuit. À ce stade, il est facile de voir que la destination d^k d'une demande $k \in K$ suit immédiatement l'origine o^k dans tous les circuits élémentaires. On identifie alors l'origine o^k et la destination d^k en un sommet k pour chaque demande $k \in K$. On obtient, pour la solution de la figure 6.4, le codage donné par la figure 6.6. Dans cette figure, comme dans les suivantes, les demandes sont représentées par des carrés dont le numéro correspond au numéro de la demande. Une telle structure de données est appelée v_0-*arborescence ordonnée à double niveaux* (v_0-*AODN*). Ce nom introduit cependant une ambiguïté qu'il est nécessaire de lever. En effet, le terme d'arborescence à double niveaux signifie ici qu'étant donnés deux sommets de l'arborescence appartenant ensemble à V ou à K, aucun des deux ne peut être fils de l'autre, même si un arc les relie. Autrement dit, les fils d'un sommet de V (respectivement K) sont des sommets de K (respectivement V).

Pour chaque demande $k \in K$, les fils de k représentent les sommets différents de o^k et d^k appartenant au chemin L_k. Pour tout sommet de $v \in V$ de la v_0-arborescence ordonnée à double niveaux, les fils de v représentent les demandes transportées sur le circuit élémentaire ayant v comme sommet d'origine. (Si v n'est pas un sommet d'origine, alors v n'appartient pas à la v_0-AODN.) Les arcs entre deux fils d'un même sommet indiquent l'ordre dans lequel ces fils sont traversés, d'où le terme "ordonnée" dans la dénomination de la structure de données. Par la suite, on note par $T = (H, B)$ le graphe représentant la v_0-arborescence ordonnée à double niveaux, où H est constitué de l'ensemble des demandes et des sommets de V appartenant à la v_0-arborescence ordonnée à double niveaux et B l'ensemble des arcs reliant les sommets de H. Une v_0-AODN est dite *couvrante* si elle couvre toutes les demandes de K, *i.e.*, $K \subseteq H$. Pour tout sommet u de H, on note par n_u le nombre de fils de u et par $F(u) = (u^1, u^2, \ldots, u^{n_u})$ la séquence ordonnée des fils de u. On remarque que pour tout sommet v de $H \cap V$, $F(v)$ est différent de l'ensemble vide puisque v est le sommet d'origine d'un circuit élémentaire transportant au moins une demande. Afin de clarifier la notion de fils d'un sommet de T, on donne les fils de chaque sommet de la v_0-AODN donnée par la figure 6.6. On a alors $F(v_0) = (1)$, $F(1) = (v_1, o^4)$, $F(v_1) = (2, 3)$, $F(o^4) = (4)$ et $F(k) = \emptyset$ pour $k = 2, 3, 4$.

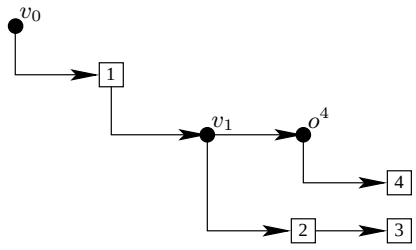

FIGURE 6.6 – v_0-arborescence ordonnée à double niveaux représentant la SOCM donnée par la figure 6.4

Il est facile de voir la relation entre solutions optimales de cardinalité minimale et v_0-arborescences ordonnées à double niveaux. On précise maintenant comment calculer le coût associé à une v_0-arborescence ordonnée à double niveaux, ce coût étant égal à celui de la SOCM représentée par la v_0-AODN. Pour cela, on associe à tout sommet de $T = (H, B)$ un coût défini de la manière suivante. Soit $v \in V \cap H$. Le coût associé à v, noté c_v, est alors égal à

$$c_v = c_{(v,o^{v^1})} + \sum_{i=1}^{n_v-1} c_{(d^{v^i},o^{v^{i+1}})} + c_{(d^{v^{n_v}},v)}.$$

Soit k un sommet de $K \cap T$. Le coût associé à k, noté c_k, est défini selon

$$c_k = \begin{cases} c_{(o^k,d^k)} & \text{si } F(k) = \emptyset \\ c_{(o^k,k^1)} + \sum_{i=1}^{n_k-1} c_{(k^i,k^{i+1})} + c_{(k^{n_k},d^k)} & \text{sinon.} \end{cases}$$

Le coût associé à une v_0-arborescence ordonnée est alors donné par

$$c(T,X) = \sum_{v \in V \cap T} c_v + \sum_{k \in K \cap T} c_k.$$

L'intérêt d'utiliser les v_0-arborescences ordonnées à double niveaux pour représenter les solutions optimales de cardinalité minimale est multiple. Premièrement, le nombre de sommets est plus petit par rapport aux codages des séquences de sommets traversés par le circuit, donnant ainsi une représentation plus compacte de la solution. De plus, cette structure de données permet de ne représenter que les solutions optimales de cardinalité minimale, contrairement à la séquence des sommets traversés par le véhicule qui permet notamment de représenter des solutions dont l'origine et la destination d'une demande n'appartiennent pas à un même circuit élémentaire de D_C. Dans une approche algorithmique telle qu'une métaheuristique, cela permet de limiter l'espace de recherche aux seules solutions intéressantes et d'accélérer ainsi tout algorithme de recherche locale. Cela évite également d'utiliser des procédures visant à vérifier que les solutions considérées respectent les propriétés des solutions recherchées.

Dans le reste du chapitre, on présente des procédures d'insertion ainsi que des opérateurs de voisinage qui seront respectivement utilisés dans l'heuristique d'insertion donnant une solution réalisable pour le 1-PRLPA unitaire métrique et dans la métaheuristique. Ces procédures utilisent les opérations d'insertion et de déplacement de sous-liste dans une liste. On explique ici de manière formelle ces opérations. Soient u un sommet de H et $l = (l^1, l^2, \ldots, l^{n_l})$ une liste d'éléments de taille n_l telle que si u est un sommet de V (respectivement K), alors l est composée d'éléments de K (respectivement V). Soit i un entier de l'ensemble $\{1, 2, \ldots, n_u\}$. Si aucun élément de l appartient à $F(u)$, alors insérer l dans $F(u)$ à la position i est l'opération modifiant $F(u)$ selon $F(u) = (u^1, u^2, \ldots, u^{i-1}, l^1, l^2, \ldots, l^{n_l}, u^i, u^{i+1}, \ldots, u^{n_u})$. Si l est une sous-liste de $F(u)$ et u^i n'est pas un élément de l, alors déplacer l d'une liste L dans la liste $F(u)$ à la position i consiste alors à supprimer tous les éléments de l dans L et à insérer l dans $F(u)$ à la position i. Par ailleurs, si $l = (l^1, l^2, \ldots, l^{n_l})$ est une sous-liste de $F(u)$, $u \in H$, on note par $\text{Prec}(l)$ (respectivement $\text{Succ}(l)$) l'élément précédent l^1 (respectivement suivant l^{n_l}) dans $F(u)$.

6.3.2 Solution initiale

Dans cette partie, on donne une heuristique permettant de construire une v_0-arborescence ordonnée T couvrant K. Cette heuristique, appelée *heuristique d'insertion*, a pour but de donner une solution initiale pour la métaheuristique développée dans la partie suivante. Cette heuristique consiste, à partir d'une v_0-AODN vide, à construire la solution en insérant itérativement toutes les demandes k de K dans la v_0-AODN. Elle utilise pour cela deux procédures d'insertion. La première, décrite par l'algorithme 10 et appelée *insertion simple*, insère la demande k comme fils d'un sommet de V appartenant à H. Cela consiste, dans le graphe D_C induit par les arcs du circuit du véhicule, à modifier un circuit élémentaire afin d'ajouter l'arc (o^k, d^k), correspondant au chemin de la demande k, dans ce circuit.

Algorithme 10: Insertion simple

Entrée : - Demande k à insérer
- Sommet $v \in H \cap V$
- Entier $i \in \{1, 2, \ldots, n_v + 1\}$

Début
Insérer k à la position i dans $F(v)$;
Fin

Un exemple d'utilisation de la procédure d'insertion simple est donné par la figure 6.7. Dans cet exemple, la demande 5 est insérée dans la v_0-AODN donnée par la figure 6.6. Les paramètres de l'algorithme d'insertion simple sont $k = 5$, $v = v_0$ et $i = 1$.

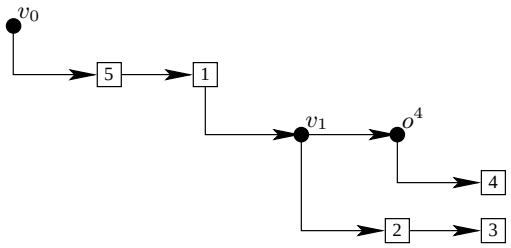

FIGURE 6.7 – Insertion simple de la demande 5 dans $F(v_0)$ à la position 1

La deuxième procédure, appelée *insertion sommet/demande*, prend en paramètre, en plus de la demande k à insérer, un sommet v de $V \setminus H$ qu'elle insère dans H comme fils d'une demande k' également donnée en paramètre. Cette procédure insère ensuite la demande k comme fils de v. Cette dernière procédure consiste, dans le graphe D_C, à

créer un nouveau circuit élémentaire, ayant v comme sommet d'origine et contenant les arcs (v, o^k), (o^k, d^k) et (d^k, v) la demande k' étant rechargée au sommet v. L'algorithme 11 décrit la procédure d'insertion sommet/demande. Ces deux procédures d'insertion ont une complexité en temps constant car les fils sont codés à l'aide d'une liste chaînée. (Il suffit de remplacer l'entier i par un pointeur sur l'élément u^{i-1}. Cependant, afin de rester clair, on garde l'entier plutôt que le pointeur.)

Algorithme 11: Insertion sommet/demande

Entrée : - Demande k à insérer
- Sommet $v \in V \setminus H$
- Demande $k' \in K \cap H$
- Entier $i \in \{1, 2, \ldots, n_{k'} + 1\}$

Début

Insérer v à la position i dans $F(k')$;
Insérer k à la position 1 dans $F(v)$;

Fin

La figure 6.8 montre l'insertion sommet/demande de la demande 6 dans la v_0-AODN donnée par la figure 6.7. Les paramètres de l'insertion sont les suivants : $k = 6$, $v = v_2$, $k' = 1$ et $i = 2$.

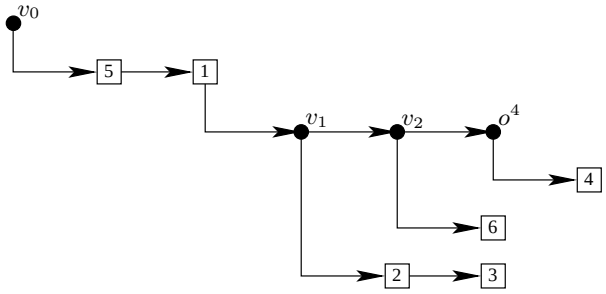

FIGURE 6.8 – Insertion sommet/demande de la demande 6 avec insertion du sommet v_2 dans $F(1)$ à la position 2

L'heuristique d'insertion détermine aléatoirement un ordre linéaire sur les demandes de K puis insère successivement chaque demande au coût minimum dans la v_0-AODN. Toutes les insertions possibles, parmi les insertions simples et les insertions sommet/demande, sont testées et celle minimisant le coût est effectuée. Déterminons maintenant la complexité de l'heuristique d'insertion construisant la solution initiale. Pour cela, déterminons la complexité de l'insertion d'une demande $k \in K \setminus H$ dans la v_0-AODN. Il faut alors tester toutes les insertions possibles pour les deux procédures d'insertion.

Pour l'insertion simple, il faut tester l'insertion de la demande k pour tout sommet $v \in V \cap H$ et pour tout entier $i \in \{1, 2, \ldots, n_v + 1\}$. Comme les fils de tout sommet $v \in V \cap H$ sont des sommets de K et que chaque sommet est fils d'un autre sommet au plus, le nombre d'applications possibles pour l'insertion simple est de l'ordre du nombre de demandes appartenant à H. Le calcul de la meilleure insertion simple se fait donc en $O(p)$. On détermine maintenant la complexité de l'insertion sommet/demande de coût minimum. Il faut tester l'algorithme d'insertion pour tout sommet de $V \setminus H$, pour toute demande $k' \in K \cap H$ et pour tout entier $i \in \{1, 2, \ldots, n_{k'} + 1\}$. Comme les fils d'un sommet de $K \cap H$ sont des sommets de V, cela revient approximativement à tester l'algorithme pour tout sommet de $V \setminus H$ et pour tout sommet de $V \cap H$ différent de v_0. La complexité de l'insertion sommet/demande au coût minimum est donc en $O(n^2)$. Comme $k \leq n$, l'insertion de coût minimum de la demande k se fait en $O(n^2)$. L'heuristique d'insertion consistant à insérer successivement les demandes selon l'ordre linéaire aléatoire, la complexité de cette heuristique est en $O(n^2 p)$.

Bien que la complexité de cette heuristique soit polynomiale, le temps d'exécution peut devenir relativement long si la cardinalité de l'ensemble V est importante. Dans ce cas, il peut être intéressant de diminuer la complexité de l'heuristique d'insertion en ne testant pas l'algorithme d'insertion sommet/demande pour tous les cas possibles. (Il n'est pas utile de diminuer le nombre de tests de l'application de l'insertion simple puisque la complexité de la détermination de l'insertion simple au coût minimum est faible.) Pour cela, une idée élémentaire consiste à tester l'insertion sommet/demande seulement pour des sommets de $V \setminus H$ "intéressants". Cette notion de sommet intéressant reste cependant à définir. L'insertion sommet/demande correspond, dans le graphe D_C, à remplacer un arc (u, v) (reliant deux sommets consécutifs dans le chemin d'une demande) par deux arcs (u, w) et (w, v), où w est le sommet de $V \setminus H$ passé en paramètre de l'algorithme d'insertion, puis à ajouter le circuit composé des trois arcs (w, o^k), (o^k, d^k) et (d^k, w). Les sommets intéressants sont donc les sommets pour lesquels le coût des quatre arcs (u, w), (w, v), (d^k, w) et (w, o^k) est le plus faible. Il est donc possible, avant de commencer l'heuristique d'insertion, de déterminer le sommet w minimisant le coût des quatre arcs pour toute paire de sommets distincts $u \neq v \in V$ et pour toute demande K. Cependant, ce calcul se fait en $O(n^3 p)$, ce qui implique que la complexité de l'algorithme d'insertion augmente. On utilise dans notre heuristique d'insertion une autre façon de calculer ces sommets dits intéressants. On définit, pour tout couple de sommets $u \neq v$ de V, le *milieu* de (u, v), noté $m(u, v)$, correspondant à $m(u, v) = \text{argmin}\{c_{(u,w)} + c_{(w,v)} : w \in V \setminus \{u, v\}\}$. Le sommet considéré comme intéressant est le milieu des milieux de u, v et d^k, o^k, *i.e.*,

$$w = m(m(u, v), m(d^k, o^k)).$$

Il est possible que le sommet w ainsi calculé ne soit pas le sommet le plus intéressant, *i.e*,

celui donnant l'insertion sommet/demande de coût minimum. De plus, il est possible que ce sommet appartienne déjà à H, impliquant que l'insertion n'est pas possible. Il faut donc sélectionner plusieurs sommets intéressants. Pour cela, on introduit la notion de *voisin* d'un sommet. Pour chaque sommet u de V, l'ensemble des voisins de u, noté $N(u)$, correspond aux R sommets de $v \in V \setminus \{u\}$ minimisant la quantité $c_{(u,v)} + c_{(v,u)}$, le nombre R définissant la cardinalité de $N(v)$ étant un paramètre de l'heuristique.

On filtre alors l'algorithme de recherche de la meilleure insertion sommet/demande en testant uniquement, pour toute demande $k' \in K \cap H$ et pour tout entier $i \in \{1, 2, \ldots, n_{k'}+1\}$, la procédure d'insertion pour les sommets de $N(m(m(u,v), m(d^k, o^k)))$. En choisissant N de manière à avoir un nombre faible de sommets (nombre constant), on diminue alors la complexité de l'heuristique d'insertion à $O(np)$.

La solution fournie par l'heuristique d'insertion dépend fortement de l'ordre dans lequel les demandes sont considérées. Il est donc intéressant d'appliquer cette heuristique pour différents ordres des demandes. Bien évidemment, il n'est pas possible d'appliquer cette heuristique pour tous les ordres possibles puisque le nombre d'ordres différents est exponentiel par rapport au nombre de demandes. Par ailleurs, appliquer l'heuristique d'insertion pour tous les ordres n'assure pas de trouver une solution optimale du problème. Un compromis consiste à appliquer un nombre constant de fois, disons Δ, l'heuristique d'insertion avec différents ordres aléatoires pour les demandes. La solution retournée, correspondant à la solution initiale, est la meilleure solution parmi les Δ solutions trouvées.

La section suivante est consacrée à la description de la métaheuristique utilisée pour améliorer la solution obtenue par l'heuristique d'insertion.

6.3.3 Métaheuristique

On décrit dans cette partie la métaheuristique utilisée pour résoudre le 1-PRLPA unitaire métrique. L'algorithme de recherche locale est une descente simple. Il part d'une solution initiale, fournie par l'heuristique d'insertion, et améliore itérativement celle-ci par modification locale de la solution. L'heuristique s'arrête lorsque plus aucune amélioration n'est possible. À chaque itération, l'algorithme tente d'améliorer la solution T, appelée *solution courante*, en cherchant une solution de coût inférieur, appelée *solution améliorante* parmi le voisinage de T. Ce voisinage sera défini après. La taille de ce dernier dépend d'un paramètre de la métaheuristique, noté H. Si aucune solution voisine améliorante n'est trouvée, alors le paramètre H est modifié afin d'étendre le voisinage de cette solution. Ceci permet un compromis entre la taille du voisinage de

la solution, permettant d'atteindre potentiellement des solutions de coût inférieur, et le temps d'exécution de l'heuristique. Cette dernière s'arrête lorsque le critère H atteint une valeur donnée. L'algorithme 12 décrit de manière plus formelle cette recherche locale. La procédure permettant d'obtenir une solution voisine est discutée par la suite.

Algorithme 12: Descente Simple

Entrée : - Instance (D, v_0, K, c) du 1-PRLPA unitaire métrique
- Paramètres H, H_{min}

Sortie : - v_0-arborescence ordonnée à double niveaux T de (D, v_0, K, c) couvrant K

Début
  T = Heuristique d'insertion;
  tant que $H \geq H_{min}$ **faire**
    T' = Solution voisine de T;
    si $c(T') < c(T)$ **alors**
      $T = T'$;
    sinon
      $H = H/2$;
  retourner T;
Fin

On définit maintenant le voisinage d'une solution T. Ce voisinage, noté $P(T)$, correspond à l'union des voisinages obtenus par application d'un opérateur de voisinage pour un ensemble de valeurs. Dès qu'une application d'un opérateur permet d'obtenir une solution améliorante, celui-ci est appliqué et la solution voisine obtenue est retournée. On ne cherche donc pas la solution de meilleur coût parmi le voisinage de T mais uniquement une solution de coût inférieur à T. L'algorithme 13 décrit la recherche d'une solution voisine améliorante en fonction des voisinages des opérateurs. On note par $P_X(T)$ le voisinage de la solution T obtenue par application de l'opérateur X. Ce voisinage correspond aux solutions obtenues par application de l'opérateur de voisinage X sur la solution T pour certaines valeurs de paramètres. Pour certains opérateurs, le voisinage dépend du paramètre H de la métaheuristique, ce qui permet de modifier le voisinage d'un opérateur durant l'algorithme. Dans ce qui suit, on présente les différents opérateurs, au nombre de six, ainsi que leur voisinage. Pour décrire ces opérateurs, il est nécessaire d'introduire la définition suivante. Soient deux sommets distincts u et v de H. On dit que u *domine* v, ou v *est dominé par* u, si, par plusieurs applications à partir du sommet v, de l'opérateur $p(x)$ retournant le sommet père du sommet x, on obtient le sommet u.

Algorithme 13: Solution voisine

Entrée : - v_0-AODN T
Sortie : - Une v_0-AODN T' voisine de T et de coût inférieur à T si elle existe, T sinon
Début
  pour *chaque opérateur* X **faire**
    $T' =$ Solution améliorante de $P_X(T)$;
    si $c(T') < c(T)$ **alors**
      retourner T';
  retourner T;
Fin

6.3.3.1 Opérateur Change Sommet

L'opérateur *Change Sommet (CS)* a pour but de remplacer le sommet d'origine d'un circuit élémentaire, disons v, par un sommet v'. Dans la v_0-arborescence ordonnée à double niveaux, cela consiste à intervertir le sommet v de $V \cap H$ avec le sommet v' de $V \setminus H$. L'algorithme de cet opérateur est celui donné par l'algorithme 14. Aucun exemple n'est donné étant donné la simplicité de cet opérateur.

Algorithme 14: Opérateur Change Sommet (CS)

Entrée : - Sommet $v \in V \cap H$
- Sommet $v' \in V \setminus H$
Début
  Remplacer le sommet v par v' dans H;
Fin

Le voisinage de l'opérateur CS d'une solution T, noté $P_{\mathrm{CS}}(T)$, correspond à l'ensemble des solutions pouvant être obtenues par application de l'opérateur CS pour tout sommet v de $V \cap H$ et pour tout sommet de $V \setminus H$ appartenant au voisinage $N(v)$ de v. (Rappelons que le voisinage $N(v)$ d'un sommet v, défini dans la section précédente, correspond à l'ensemble des sommets (de taille fixée par un paramètre) w minimisant la somme des coûts des arcs (v, w) et (w, v).)

6.3.3.2 Opérateur 3 Inter-Échange Sommet

L'opérateur *3 Inter-Échange Sommet (3-IES)* consiste à modifier l'ordre des sommets de rechargements dans le chemin d'une demande. Ceci correspond, dans la v_0-AODN,

à modifier la séquence des fils d'une demande. Cette modification de l'ordre des fils se fait par déplacement d'une sous-liste des fils dans la liste. On parle alors de 3 Inter-Échange car, si l'on identifie l'origine et la destination de la demande, on obtient un circuit élémentaire. De plus, le déplacement de la sous-liste revient, dans ce circuit élémentaire, à remplacer trois arcs de ce circuit (l'arc précédent le premier élément de la sous-liste, l'arc suivant le dernier élément et celui entre les deux sommets où l'on souhaite introduire la sous-liste) par trois autres arcs. L'algorithme 15 décrit cet opérateur.

Algorithme 15: Opérateur 3 Inter-Échange Sommet (3-IES)

Entrée : - Demande $k \in K$
- Sous-liste l de $F(k)$
- Entier $i \in \{1, 2, \ldots, n_k + 1\}$ tel que $k^i \notin l$

Début

Déplacer l à la position i dans $F(k)$;

Fin

L'unique condition d'application de l'opérateur 3-IES est que la position à laquelle on insère la sous-liste dans la liste des fils ne soit pas comprise dans la sous-liste. Dans la figure 6.9, on donne un exemple d'application de l'opérateur 3-IES sur la v_0-AODN de la figure 6.8. Les paramètres d'entrée sont les suivants : $k = 1$, $l = (v_2, o^4)$ et $i = 1$.

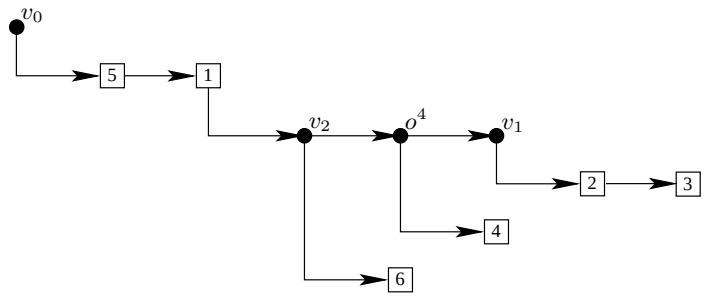

FIGURE 6.9 – Application de l'opérateur 3-IES

Le voisinage $P_{3-\mathrm{IES}}(T)$ de l'opérateur 3 Inter-Échange Sommet pour une solution T donnée est l'ensemble des solutions obtenues par application de cet opérateur pour toute demande k de K, pour toute sous-liste $l = (l^1, l^2, \ldots, l^{n_l})$ telle que les coûts des arcs $(\mathrm{Prec}(l), l^1)$ et $(l^{n_l}, \mathrm{Succ}(l))$ sont supérieurs ou égaux à H, et pour tout entier i tel que $c(k^{i-1}, k^i) \geq H$. Il est clair que si l^1 (respectivement l^{n_l}) est le premier (respectivement dernier) fils de k, alors l'arc $(\mathrm{Prec}(l), l^1)$ (respectivement $(l^{n_l}, \mathrm{Succ}(l))$) correspond à l'arc (o^k, k^1) (respectivement (k^{n_k}, d^k)). De même, si $i = 0$ ou si $i = n_k + 1$, alors l'arc

(k^{i-1}, k^i) correspond à l'arc (o^k, k^1) ou (k^{n_k}, d^k) respectivement. Dans la description des voisinages des autres opérateurs, on ne discutera pas de ces cas particuliers puisque ces derniers sont similaires.

6.3.3.3 Opérateur 3 Inter-Échange Sommet Externe

L'opérateur *3 Inter-Échange Sommet Externe (3-IES Ext)* consiste à changer la demande passant par certains sommets de rechargement. De manière plus précise, l'opérateur prend en entrée une séquence de sommets de rechargements appartenant au chemin d'une demande $k \in K$. Il prend également en entrée une deuxième demande k' différente de k et modifie la v_0-AODN de manière à ce que la séquence de rechargements appartienne désormais au chemin de la demande k', la position dans $L_{k'}$ étant donnée par l'entier i. Ceci revient finalement à déplacer la séquence dans la liste des fils de k'. L'algorithme de cet opérateur est donné dans l'algorithme 16.

Algorithme 16: Opérateur 3 Inter-Échange Sommet Externe (3-IES Ext)

Entrée : - Demandes $k \neq k' \in K$
- Sous-liste l de $F(k)$ telle que k' n'est pas dominée par un élément de l
- Entier $i \in \{1, 2, \ldots, n_{k'} + 1\}$

Début

Déplacer l à la position i dans $F(k')$;

Fin

La condition pour que l'opérateur 3-IES Ext soit appliquable est que k' ne soit dominée par aucun élément de l car après application de cet opérateur, tous les éléments de l vont être fils de k' et donc dominés par k'. Or, comme on a une v_0-arborescence ordonnée à double niveaux, deux sommets ne peuvent se dominer mutuellement. Dans la figure 6.10, on donne un exemple d'application de l'opérateur 3-IES Ext sur la v_0-AODN de la figure 6.8 lorsque $k = 1$, $k' = 2$, $l = (v_2, o^4)$ et $i = 1$.

Le voisinage $P_{3-\text{IESExt}}(T)$ de l'opérateur 3 Inter-Échange Sommet Externe pour une solution T donnée est l'ensemble des solutions obtenues par application de cet opérateur pour toute paire de demandes distinctes $k \neq k'$ de K, pour toute sous-liste $l = (l^1, l^2, \ldots, l^{n_l})$ telle que $c(\text{Prec}(l), l^1) \geq H$ et $c(l^{n_l}, \text{Succ}(l)) \geq H$ et pour tout entier i tel que $c(k'^{i-1}, k'^i) \geq H$.

Les deux opérateurs précédents modifient la séquence des fils des demandes. Il est également possible d'appliquer le même type d'opérateurs pour modifier la séquence

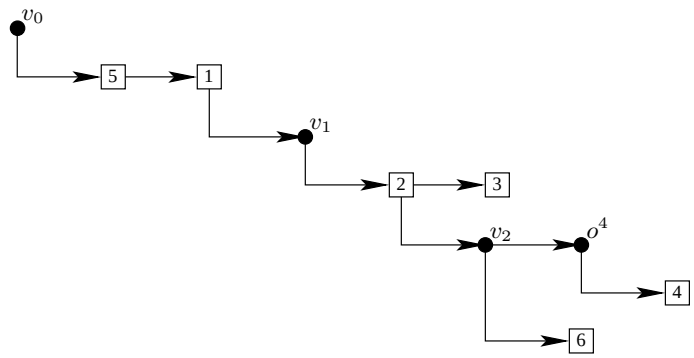

FIGURE 6.10 – Application de l'opérateur 3-IES Ext

des fils des sommets de rechargements, *i.e*, des sommets de $V \cap H$. Ceci est fait avec les deux opérateurs suivants.

6.3.3.4 Opérateur 3 Inter-Échange Demande

L'opérateur *3 Inter-Échange Demande (3-IED)* modifie l'ordre des fils d'un même sommet v de $V \cap H$. Cette modification correspond à une application d'un 3 Inter-Échange dans le circuit élémentaire ayant v comme sommet d'origine dans le graphe D_C. L'algorithme 17 décrit cet opérateur.

Algorithme 17: Opérateur 3 Inter-Échange Demande (3-IED)

Entrée : - Sommet $v \in V \cap H$
- Sous-liste l de $F(v)$
- Entier $i \in \{1, 2, \ldots, n_{k'} + 1\}$ tel que $v^i \notin l$

Début

Déplacer l à la position i dans $F(v)$;

Fin

On donne un exemple d'application de l'opérateur 3-IED sur la v_0-arborescence ordonnée à double niveaux donnée par la figure 6.8. Les paramètres d'entrée sont les suivants : $v = v_1$, $l = (2)$ et $i = 2$.

Le voisinage $P_{3-\text{IED}}(T)$ correspond aux solutions obtenues à partir de T par application de l'opérateur 3-IED pour tout sommet v de $V \cap H$, pour toute sous-liste $l = (l^1, l^2, \ldots, l^{n_l})$ telle que $c(d^{\text{Prec}(l)}, o^{l^1}) \geq H$ et $c(d^{l^{n_l}}, o^{\text{Succ}(l)}) \geq H$, et pour tout entier i tel que $c(d^{v^{i-1}}, o^{v^i}) \geq H$.

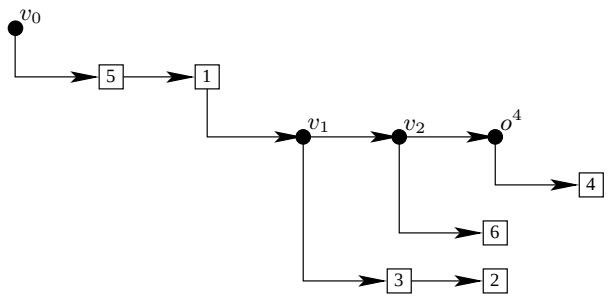

FIGURE 6.11 – Application de l'opérateur 3-IED

6.3.3.5 Opérateur 3 Inter-Échange Demande Externe

L'opérateur *3 Inter-Échange Demande Externe (3-IED Ext)* est similaire à l'opérateur 3-IES Ext sauf qu'il s'applique aux fils d'un sommet de $V \cap H$. Ceci correspond, dans D_C, à modifier le circuit élémentaire transportant les demandes de l. L'algorithme 18 décrit cet opérateur.

Algorithme 18: Opérateur 3 Inter-Échange Demande Externe (3-IED Ext)

Entrée : - Sommets $v \neq v' \in V \cap H$
- Sous-liste l de $F(v)$ telle que v' n'est pas dominé par un élément de l
- Entier $i \in \{1, 2, \ldots, n_{v'} + 1\}$

Début

Déplacer l à la position i dans $F(v')$;
si $F(v) = \emptyset$ **alors**
Supprimer v dans H;

Fin

Contrairement à l'opérateur 3-IES Ext, il faut vérifier, après suppression de l dans $F(v)$, si $F(v)$ est différent de l'ensemble vide. Autrement, cela implique que v est une feuille de la v_0-AODN et il faut alors supprimer v de T. En effet, la demande k dont v est le fils n'a plus d'intérêt à passer par ce sommet puisque ce dernier n'est plus un sommet de rechargement. La condition d'application de cet opérateur est similaire à celle donnée pour l'opérateur 3-IES Ext. La figure 6.12 montre l'application de l'opérateur 3-IED Ext sur la v_0-AODN de la figure 6.8 lorsque $v = v_2$, $l = (6)$, $v' = v_0$ et $i = 3$.

Le voisinage $P_{3-\text{IEDExt}}(T)$ correspond aux solutions obtenues à partir de T par application de l'opérateur 3-IED Ext pour toute paire de sommets distincts $v \neq v'$ de $V \cap H$, pour toute sous-liste $l = (l^1, l^2, \ldots, l^{n_l})$ telle que les coûts des arcs $(d^{\text{Prec}(l)}, o^{l^1})$ et

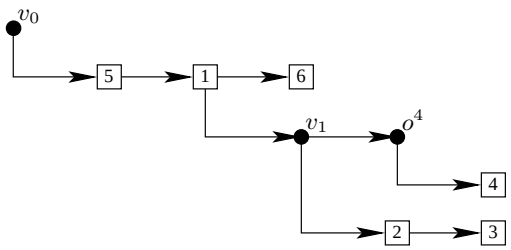

FIGURE 6.12 – Application de l'opérateur 3-IED Ext

$(d^{l^{n_l}}, o^{\mathrm{Succ}(l)})$ sont supérieurs ou égaux à H, et pour tout entier i tel que $c(d^{v'^{i-1}}, o^{v'^i}) \geq H$.

6.3.3.6 Opérateur Ajout de Sommet

Il est facile de voir que les cinq premiers opérateurs ne modifient pas le nombre de sommets de V dans T. L'opérateur 3-IED Ext peut, dans certains cas, supprimer un sommet de T. Il est donc nécessaire d'introduire un opérateur permettant d'ajouter dans T un sommet de $V \setminus H$, disons v'. Cet opérateur, appelé *Ajout de Sommet (AS)*, introduit ce sommet v' comme fils d'une demande et prend une sous-liste d'un sommet de $V \cap H$ qu'il fixe comme la séquence fils de v'. Ceci correspond, dans le graphe D_C, à prendre un sous-chemin d'un circuit élémentaire pour créer un nouveau circuit élémentaire ayant v' comme sommet d'origine. L'algorithme 19 décrit précisément l'opérateur d'ajout de sommet.

Algorithme 19: Opérateur Ajout de Sommet (AS)

Entrée : - Sommet $v \in V \cap H$
- Sous-liste l de $F(v)$
- Demande $k \in K$ non dominé par un élément de l
- Entier $i \in \{1, 2, \ldots, n_k + 1\}$
- Sommet $v' \in V \setminus H$

Début
- Insérer v' à la position i dans $F(k)$;
- Insérer l à la position 1 dans $F(v')$;
- **si** $F(v) = \emptyset$ **alors**
  - Supprimer v dans H;

Fin

La condition requise pour que cet opérateur puisse être appliqué est que la demande

k ne soit dominée par aucun élément de l car après application de cet opérateur, k domine tous les éléments de l puisque ces derniers sont fils de v' qui est lui-même fils de k. De même que pour l'opérateur 3-IED Ext, si $F(v) = \emptyset$, alors on supprime v de T afin que les feuilles de T ne soient que des sommets de K. On donne, dans la figure 6.13, la solution obtenue, à partir de celle donnée par la figure 6.8, par application de l'opérateur ajout de sommet lorsque $v = v_1$, $l = (2)$, $k = 5$, $i = 1$ et $v' = v_3$.

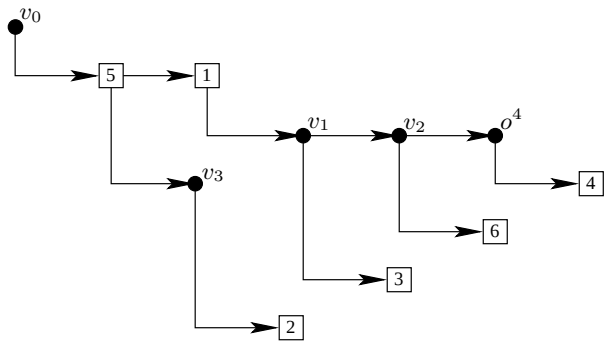

FIGURE 6.13 – Application de l'opérateur AS

Le voisinage $P_{\text{AS}}(T)$ correspond aux solutions obtenues à partir de T par application de l'opérateur AS pour tout sommet v de $V \cap H$, pour toute sous-liste $l = (l^1, l^2, \ldots, l^{n_l})$ telle que $c(d^{\text{Prec}(l)}, o^{l^1}) \geq H$ et $c(d^{l^{n_l}}, o^{\text{Succ}(l)}) \geq H$, pour toute demande k de K, pour tout entier i tel que $c(k^{i-1}, k^i) \geq H$ et pour tout sommet de $V \cap H$ appartenant au voisinage $N(v)$ de v.

Dans la section suivante, on présente les résultats expérimentaux obtenus par l'heurisitique d'insertion et la métaheuristique pour des instances du 1-PRLPA unitaire métrique.

6.3.4 Résultats expérimentaux

Dans cette partie, on décrit les instances testées et discutons des résultats expérimentaux obtenus.

6.3.4.1 Description des instances traitées

Les résultats qui sont présentés ici ont été obtenus à partir de différents types d'instances. Tout d'abord, les résultats ont été testées sur des instances résolues de manière optimale par l'algorithme de coupes et branchements présenté dans le chapitre

5. D'autres instances ont été construites de telle manière que la solution optimale soit connue et contienne beaucoup de rechargements. Cette construction permet de tester l'efficacité de l'heuristique sur des instances de grande taille. De plus, elle permet d'étudier le comportement de l'heuristique lorsque les solutions optimales contiennent beaucoup de rechargements. On présente maintenant de manière plus précise les différents types d'instances testées.

Instances issues de la TSP librairy Ces instances proviennent de la TSP Librairy [82]. Le graphe est obtenu en considérant un sous-ensemble de sommets de l'instance de la TSP Librairy. Le graphe est complet et contient entre 31 et 71 sommets dans les instances testées. Les demandes ont été générées aléatoirement de manière à ce que chaque sommet soit incident à au plus une demande, le dépôt étant incident à aucune demande. Le nombre de demandes pour chaque taille de graphe varie de 5 en 5 jusqu'au nombre maximum de demandes permis, correspondant à $\lfloor \frac{n-1}{2} \rfloor$. Les coûts des arcs du graphe peuvent être symétriques ou asymétriques. On parle alors respectivement d'instances symétriques et d'instances asymétriques.

Instances spécifiques Des instances spécifiques ont été créées de telle manière que la solution optimale contienne beaucoup de rechargements et que le coût de la solution optimale soit égal à la somme des coûts des arcs allant de l'origine à la destination de chaque demande. Les coûts de ces instances sont symétriques. Ces instances sont construites de la manière suivante. Le nombre p de demandes est un multiple de trois. Le nombre de sommets n est minimum par rapport à p, *i.e.*, $n = 2p + 1$. L'instance est définie par les coordonnées des sommets dans un plan. Ces coordonnées sont choisies de manière à ce que la solution optimale contienne $\frac{p}{3} - 1$ rechargements. Dans le graphe D_C induit par les arcs traversés par le véhicule, cela revient à avoir $\frac{p}{3}$ circuits élémentaires. Chaque circuit élémentaire C_i contient trois demandes, disons les demandes 1,2 et 3. Le sommet d'origine de C_i est le sommet o^1. De plus, les arcs (d^1, o^2), (d^2, o^3) et (d^3, o^1) de C_i sont de coût nul. Par ailleurs, le sommet d'origine v_i d'un circuit C_i, $i \neq 0$, se trouve sur le "milieu" d'une demande k transportée sur un circuit C_j, $j \neq i$, *i.e.*, $c_{(o^k,v_i)} + c_{(v_i,d^k)} = c_{(o^k,d^k)}$. Le coût de la solution optimale est alors égal à $\sum_{k \in K} c_{(o^k,d^k)}$.

6.3.4.2 Résultats expérimentaux

Avant de présenter les différents résultats expérimentaux que nous avons obtenus, nous donnons un bref descriptif du contexte informatique. Les algorithmes ont été développés en C++ et testés sur un Intel(R) Core(TM)2 Duo CPU P8400 à 2,26 GHz

avec 4Go de mémoire vive, sous système linux. Aucune limite de temps n'a été fixée. Les paramètres de la métaheuristique sont les suivants. Le paramètre R définissant la cardinalité du voisinage $N(v)$ d'un sommet v de V est égal à 4. Le paramètre Δ, correspondant au nombre d'exécutions de l'heuristique d'insertion, est égal à 100. Le paramètre H de la métaheuristique est égal à la moyenne des coûts des arcs de la solution initiale retournée par l'heuristique d'insertion. H_{min} est égal au coût de l'arc de coût minimum dans la solution initiale. Nos résultats expérimentaux sont reportés dans les tables suivantes. Les différentes colonnes représentent :

- n : le nombre de sommets du graphe,
- p : le nombre de demandes de l'instance,
- GapI1 : l'erreur relative entre le coût de la meilleure solution trouvée par l'heuristique d'insertion et celui de la solution optimale,
- RI1 : le nombre de rechargements dans la meilleure solution trouvée par l'heuristique d'insertion,
- GapI2 : l'erreur relative entre le coût de la meilleure solution trouvée par l'heuristique d'insertion et la moyenne des coûts des solutions trouvées par l'heuristique,
- RI2 : le nombre moyen de rechargements dans les solutions trouvées par l'heuristique d'insertion,
- TI : temps d'exécution de l'heuristique d'insertion en millisecondes,
- GapM : l'erreur relative entre le coût de la solution trouvée par la métaheuristique et celui de la solution optimale,
- RM : le nombre de rechargements dans la solution trouvée par la métaheuristique,
- TM : temps d'exécution de la métaheuristique une fois que la solution initiale est trouvée en secondes,
- ROpt : le nombre de rechargements dans la solution optimale.

Dans les tables suivantes, chaque ligne correspond à la moyenne des résultats obtenus en exécutant cinq instances ayant le même nombre de sommets et de demandes. Pour les instances issues de la TSP librairy, les coûts des cinq instances sont les mêmes. Seules les origines et destinations des demandes changent. Pour les instances spécifiques, les coûts varient suivant les cinq instances.

La table 6.1 présente les résultats obtenus pour les instances symétriques. Concernant l'heuristique d'insertion, on peut remarquer que le temps d'exécution est vraiment très court. Au maximum 50 millisecondes sont nécessaires pour exécuter 100 fois l'heuristique d'insertion. De plus, l'erreur relative entre la meilleure solution fournie par les 100 applications de l'heuristique et la solution optimale est très faible ; elle est toujours inférieure à 3.4%. On peut remarquer que le nombre de rechargements dans la solution optimale est généralement plus élevé que celui dans la solution initiale. Par ailleurs, il est intéressant d'appliquer plusieurs fois l'heuristique initiale. En effet, les coûts des solutions obtenues lors des différentes applications de l'heuristique peuvent varier de

manière significative. Si l'on considère les instances à 51 sommets et 10 demandes, l'erreur relative entre la meilleure solution trouvée et la solution optimale est de 0.12% alors que celle entre le coût moyen des solutions trouvées et la solution optimale est de 4.35%. On remarque finalement que la meilleure solution trouvée contient généralement plus de rechargements que la moyenne.
La métaheuristique fournit de très bons résultats. En effet, l'erreur relative par rapport à la solution optimale est au plus de 0.55%. Elle améliore presque toujours la solution retournée par l'heuristique initiale et permet de diminuer l'erreur relative de celle-ci jusqu'à près de 3%. Contrairement à l'heuristique initiale, le nombre de rechargements dans la solution de la métaheuristique est généralement plus élevé que celui

n	p	GapI1	RI1	GapI2	RI2	TI	GapM	RM	TM	ROpt
31	5	0.00	0.40	3.76	0.18	0.00	0.00	0.40	0.01	0.40
31	10	0.48	1.00	3.68	0.48	2.00	0.14	1.20	0.07	1.40
31	15	1.71	0.80	3.59	0.41	4.00	0.38	1.40	0.27	0.80
41	5	0.75	0.20	1.30	0.15	0.00	0.00	0.60	0.02	0.60
41	10	0.75	0.20	3.63	0.20	0.00	0.55	0.80	0.14	1.40
41	15	0.56	1.00	3.53	0.51	8.00	0.19	2.20	0.33	1.40
41	20	2.82	0.80	3.85	0.62	18.00	0.48	3.40	1.39	2.80
51	5	0.10	0.20	1.06	0.15	2.00	0.00	0.40	0.05	0.60
51	10	0.12	0.60	4.35	0.39	4.00	0.02	0.60	0.16	1.00
51	15	0.75	0.20	2.53	0.21	8.00	0.34	0.60	0.50	0.40
51	20	1.26	0.20	3.74	0.38	16.00	0.30	1.60	1.29	1.20
51	25	1.42	0.20	3.20	0.38	20.00	0.08	1.20	2.87	0.60
61	5	0.00	0.20	2.91	0.18	4.00	0.00	0.20	0.04	0.20
61	10	0.20	0.60	4.27	0.35	0.00	0.02	1.00	0.15	1.00
61	15	0.53	0.60	4.05	0.32	10.00	0.09	0.80	0.43	0.60
61	20	1.26	0.20	3.75	0.32	18.00	0.13	2.00	1.24	1.20
61	25	2.67	0.60	3.60	0.56	22.00	0.28	2.40	3.64	2.00
61	30	2.52	0.00	3.30	0.37	42.00	0.22	2.80	5.02	0.60
71	5	0.10	0.20	2.70	0.08	2.00	0.10	0.20	0.04	0.20
71	10	0.36	0.00	3.40	0.20	6.00	0.00	0.60	0.21	0.40
71	15	0.66	0.60	4.39	0.36	12.00	0.12	0.60	0.67	0.40
71	20	1.50	0.60	4.41	0.50	18.00	0.26	1.40	2.07	1.00
71	25	2.93	0.40	4.10	0.35	24.00	0.48	2.60	4.47	1.60
71	30	3.39	0.80	3.51	0.54	38.00	0.55	2.80	7.75	1.20
71	35	3.24	0.80	4.04	0.65	50.00	0.40	3.60	12.29	1.60

TABLE 6.1 – Résultats obtenus sur les instances aléatoires symétriques

dans la solution optimale. Finalement, le temps d'exécution de la métaheuristique est relativement faible.

n	p	GapI1	RI1	GapI2	RI2	TI	GapM	RM	TM	ROpt
31	5	0.00	0.00	1.18	0.00	0.00	0.00	0.00	0.02	0.00
31	10	0.17	0.20	2.74	0.07	0.00	0.14	0.20	0.11	1.00
31	15	1.30	0.00	3.53	0.06	8.00	0.96	0.00	0.33	1.20
41	5	0.05	0.00	0.27	0.00	0.00	0.00	0.60	0.05	0.60
41	10	0.02	0.40	0.15	0.19	0.00	0.00	0.40	0.30	0.40
41	15	0.01	0.00	0.21	0.10	6.00	0.01	0.20	0.68	0.00
41	20	0.02	0.00	0.15	0.04	20.00	0.00	0.00	1.87	0.00
51	5	0.60	0.00	0.56	0.00	0.00	0.60	0.00	0.02	0.40
51	10	0.17	0.00	3.53	0.07	2.00	0.17	0.00	0.18	0.60
51	15	0.82	0.20	3.62	0.25	6.00	0.50	0.40	0.63	1.00
51	20	2.00	0.00	3.74	0.12	16.00	1.20	0.00	1.83	1.00
51	25	2.51	0.00	3.08	0.12	20.00	0.54	0.20	3.67	1.40
61	5	1.23	0.00	3.26	0.08	4.00	0.24	0.40	0.05	0.60
61	10	0.26	0.20	4.61	0.28	2.00	0.26	0.20	0.21	0.40
61	15	1.28	0.00	3.89	0.30	10.00	0.10	0.80	0.95	0.60
61	20	1.24	0.00	4.13	0.27	16.00	0.51	0.00	1.91	0.40
61	25	2.47	0.00	4.49	0.33	20.00	0.96	1.20	3.97	0.60
61	30	3.44	0.00	3.31	0.20	34.00	1.12	1.20	6.29	0.20
71	5	0.00	0.00	0.90	0.00	2.00	0.00	0.00	0.05	0.00
71	10	0.98	0.00	0.52	0.00	0.00	0.00	0.20	0.27	0.20
71	15	0.16	0.20	1.43	0.13	8.00	0.08	0.20	0.90	0.20
71	20	0.00	0.00	1.34	0.01	18.00	0.00	0.00	2.08	0.00
71	25	0.05	0.00	1.54	0.00	20.00	0.00	0.00	4.35	0.00
71	30	0.30	0.20	1.47	0.07	38.00	0.22	0.40	7.88	0.40
71	35	0.34	0.00	1.59	0.02	54.00	0.04	0.00	13.97	0.00

TABLE 6.2 – Résultats obtenus sur les instances aléatoires asymétriques

La table 6.2 rapporte les résultats obtenus pour les instances asymétriques. La meilleure solution fournie après 100 applications de l'heuristique d'insertion est de bonne qualité. L'erreur relative par rapport à la solution optimale est inférieure à 3.5%. De même que pour les instances symétriques, le nombre de rechargements dans cette solution est inférieur ou égal à celui dans la solution optimale. Par contre, contrairement au cas symétrique, il est généralement plus faible que le nombre moyen de rechargements dans les différentes solutions retournées par l'heuristique d'insertion. Comme pour le cas symétrique, il est intéressant d'appliquer plusieurs fois l'heuristique initiale.

En effet, appliquer 100 fois cette heuristique permet d'améliorer les solutions jusqu'à plus de 4.5%. Finalement, on remarque que le temps d'exécution est négligeable.
La métaheuristique améliore presque toujours la solution initiale. Elle améliore celle-ci jusqu'à plus de 2.3%. De plus, l'application de la métaheuristique augmente généralement le nombre de rechargements. La solution fournie par la métaheuristique est de très bonne qualité. L'erreur relative entre cette solution et la solution optimale est inférieure à 1.2%. Par ailleurs, le nombre de rechargements est plus faible que celui de la solution optimale. Les temps d'exécution de la métaheuristique sont très faibles puisque pour chaque instance, ils ne dépassent pas 14 secondes.

n	p	GapI1	RI1	GapI2	RI2	TI	GapM	RM	TM	ROpt
25	12	3.80	1.80	9.64	0.19	2.00	0.37	3.00	0.09	3
37	18	5.69	2.00	10.33	0.24	10.00	0.00	5.00	0.22	5
49	24	10.84	1.60	7.96	0.33	20.00	0.79	7.00	0.65	7
61	30	10.22	1.20	7.51	0.44	30.00	0.15	9.00	1.42	9
73	36	10.71	1.20	7.19	0.37	50.00	1.07	11.00	2.95	11
85	42	11.50	0.80	6.88	0.43	74.00	0.46	13.00	5.31	13
97	48	11.04	1.20	6.18	0.60	88.00	0.89	15.00	8.49	15
109	54	11.23	1.20	5.95	0.64	126.00	1.56	17.00	14.74	17
121	60	11.50	1.20	6.26	0.64	156.00	0.52	19.00	20.27	19
133	66	12.21	1.80	6.12	0.73	166.00	1.42	21.00	29.72	21
145	72	12.57	1.40	5.74	0.78	206.00	1.39	22.80	51.52	23
157	78	13.82	1.20	5.30	0.81	234.00	1.37	25.00	60.19	25
169	84	14.19	1.00	4.80	0.90	274.00	0.78	26.80	80.22	27
181	90	14.53	1.80	5.22	1.00	328.00	1.42	28.80	108.10	29
193	96	13.81	1.00	4.88	1.07	368.00	1.26	30.80	137.98	31
205	102	14.19	1.00	4.67	0.92	408.00	1.10	32.80	198.78	33
217	108	14.40	2.20	4.23	1.05	444.00	0.75	35.00	247.67	35
229	114	14.76	0.80	4.11	1.09	512.00	1.01	36.60	312.14	37
241	120	14.25	1.60	4.39	1.09	566.00	1.01	38.80	387.88	39
253	126	14.39	2.00	4.53	1.38	622.00	1.06	40.80	455.55	41
265	132	14.84	2.40	4.28	1.33	676.00	1.11	42.80	572.23	43
277	138	14.86	1.80	3.78	1.34	730.00	0.70	44.60	639.81	45
289	144	14.90	2.00	4.23	1.24	796.00	1.15	47.00	790.75	47
301	150	14.98	1.20	3.92	1.13	870.00	1.34	49.00	897.31	49

TABLE 6.3 – Résultats obtenus sur les instances spécifiques

La table 6.3 donne les résultats expérimentaux obtenus pour les instances spécifiques. On remarque que la solution fournie par l'heuristique d'insertion n'est pas de très bonne

qualité. En effet, l'erreur relative entre la meilleure solution obtenue par cette heuristique et la solution optimale augmente en fonction de la taille des instances et atteint près de 15% pour les instances à 301 sommets et 150 demandes. De plus, pour les instances ayant au moins 49 sommets et 24 demandes, elle est au minimum de 10%. Ceci s'explique partiellement par le faible nombre de rechargements dans la solution intiale par rapport à celui dans la solution optimale. Alors que ce dernier atteint 49 pour les instances à 301 sommets et 150 demandes, la solution initiale ne contient jamais plus de 2.4 rechargements en moyenne. Pour ces instances, il est très important de lancer plusieurs fois l'heuristique initiale. En effet, l'erreur relative entre le coût moyen et le coût de la meilleure solution varie jusqu'à dépasser 10%. Si les solutions obtenues par cette heuristique ne sont pas très satisfaisants, le temps d'exécution est toujours très rapide. Les 100 exécutions ne dépassent jamais une seconde.
La métaheuristique joue ici un rôle important. Elle permet en effet d'améliorer considérablement la qualité de la solution obtenue par l'heuristique d'insertion. La solution donnée par la métaheuristique est de bonne qualité ; l'erreur relative est inférieure à 1.6% par rapport à la solution optimale. On remarque que le nombre de rechargements est quasi identique à celui de la solution optimale. Le temps d'exécution de cette métaheuristique reste relativement faible - moins de 15 minutes - au vu de la taille des instances et de la qualité des solutions obtenues.

Les résultats expérimentaux permettent de mesurer la qualité de l'approche heuristique développée dans ce chapitre pour résoudre le 1-PRLPA unitaire métrique. L'heuristique d'insertion permet d'obtenir rapidement une solution réalisable. Cependant, cette heuristique montre rapidement ses limites. En effet, la qualité de la solution obtenue dépend fortement du nombre de rechargements dans la solution optimale. Plus ce nombre augmente, plus la qualité de la solution obtenue se dégrade. Il semble très difficile, pour cet algorithme, d'appréhender la complexité induite par les rechargements. De plus, cette heuristique dépend beaucoup de l'ordre des demandes considéré. Il est donc à craindre que la qualité de la solution diminue lorsque la taille des instances augmente, le nombre d'ordres possibles augmentant exponentiellement par rapport au nombre de demandes. Cela justifie aussi notre choix d'appliquer plusieurs fois l'heuristique d'insertion pour chaque instance.
La métaheuristique fournit des solutions de très bonne qualité en un temps acceptable. Elle améliore quasiment toujours la solution initiale donnée en paramètre. Cette amélioration se fait notamment par l'introduction de rechargements dans la solution initiale. En effet, le nombre de rechargements dans la solution après application de la métaheuristique est supérieur ou égal au nombre de rechargements dans la solution initiale. La métaheuristique comble ainsi la faiblesse de l'heuristique d'insertion qui peine à traiter efficacement les rechargements.

Dans ce chapitre, nous nous sommes intéressés au problème de ramassage et livraison mono-véhicule préemptif asymétrique unitaire métrique. Nous avons d'abord étudié la structure des solutions de ce problème. Nous avons montré que l'information minimale nécessaire pour décrire les solutions est l'ensemble des arcs traversés par le véhicule. En effet, si l'on connait l'ensemble d'arcs traversé par le véhicule, on peut déterminer en temps polynomial si cet ensemble d'arcs correspond à une solution réalisable du problème. De plus, si tel est le cas, nous avons montré que cet ensemble d'arcs induit un cactus. Nous avons alors ramené le 1-PRLPA unitaire métrique à la recherche d'un cactus de coût minimum vérifiant certaines conditions. Nous avons ensuite formulé le problème à l'aide d'un programme linéaire en nombres entiers. Finalement, nous avons développé une métaheuristique permettant de résoudre le 1-PRLPA unitaire métrique. Cet algorithme repose sur les résultats théoriques précédents qui ont permis d'utiliser une structure de données efficace pour coder les solutions. Les résultats expérimentaux obtenus montrent ainsi la qualité de l'heuristique développée.

Conclusion

Dans cette thèse, nous avons étudié différentes variantes du problème de ramassage et livraison préemptif.

Dans un premier temps, nous avons considéré le problème de ramassage et livraison préemptif lorsque la flotte est composée de plusieurs véhicules et les demandes peuvent être transportées avec fractionnement. Pour cette variante, nous avons modélisé le problème à l'aide d'un graphe spatio-temporel puis nous avons formulé le problème à l'aide de deux programmes linéaires en nombres entiers basés sur cette modélisation. Pour chaque formulation, nous avons développé un algorithme de coupes et branchements. Nous avons finalement comparé les résultats expérimentaux obtenus pour chaque algorithme.

Nous nous sommes ensuite intéressés à un cas particulier de ce problème dans lequel un seul véhicule est disponible et le fractionnement des demandes n'est pas autorisé. Nous avons montré que toute solution doit être décrite par l'ensemble des arcs traversés par le véhicule, l'ordre dans lequel ces arcs sont traversés et les arcs empruntés par chaque demande. En effet, on a prouvé que si l'une de ces informations manque, alors déterminer si une solution est réalisable est un problème NP-complet. Nous avons également montré que si le véhicule transporte une seule demande à la fois, alors l'ordre sur les arcs du véhicule n'est plus nécessaire pour décrire une solution du problème. Pour cela, on a montré que tester si une solution, décrite par les différents ensembles d'arcs, est réalisable peut se faire dans ce cas en temps polynomial. Nous avons alors introduit deux programmes linéaires en nombres entiers différents suivant que le véhicule transporte une ou plusieurs demandes à la fois. Pour la première formulation (cas unitaire), nous avons mené une étude polyédrale. Nous avons donné des conditions nécessaires et suffisantes pour que les contraintes de la formulation définissent des facettes. Nous avons ensuite développé un algorithme de coupes et branchements basé sur ces résultats. Finalement, nous avons décrit le polytope du problème lorsque le graphe est un cactus et l'ensemble des demandes vérifie certaines conditions.

Nous avons considéré ensuite la version unitaire du problème lorsque l'instance vérifie les hypothèses suivantes : le graphe est complet, chaque sommet est incident à au plus une demande et les coûts vérifient les inégalités triangulaires. Nous avons montré que, dans ce cas, une solution pouvait être décrite par l'ensemble des arcs traversés par le véhicule. Nous avons aussi montré que ce problème se ramène à la recherche d'un cactus de coût minimum vérifiant certaines propriétés. Une métaheuristique, basée sur cette transformation, a été développée pour le problème dans ce cas.

Ce travail ouvre plusieurs perspectives, aussi bien d'un point de vue polyédral qu'algorithmique. Tout d'abord, sur le plan polyédral, l'étude menée pourrait être approfondie en identifiant de nouvelles contraintes valides. Celles-ci pourraient permettre d'améliorer l'algorithme de coupes et branchements proposé dans la thèse. De plus, il serait intéressant d'étendre la description polyédrale à d'autres classes de graphes. D'un point de vue algorithmique, il serait utile d'améliorer les algorithmes de séparation des différentes classes de contraintes. Par ailleurs, comme il a été remarqué, une grande partie du temps d'exécution est utilisée dans la résolution de la relaxation linéaire initiale. Ceci est dû au fait que le nombre de variables devient très grand, même pour des instances de taille moyenne. Il serait donc intéressant de considérer une approche arc-chemins pour formuler le problème. Un algorithme basé sur la génération de colonnes pourrait alors être développé et combiné avec un algorithme de coupes. Il est possible qu'une telle approche permette de résoudre des instances de taille plus importante.

Une autre perspective intéressante serait de développer un algorithme de coupes et branchements pour la variante du problème étudiée dans le chapitre 6. Cet algorithme pourrait s'appuyer sur la formulation donnée dans ce chapitre.

Bibliographie

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows : Theory, Algorithms, and Applications*. Prentice Hall, February 1993.

[2] S. Anily, M. Gendreau, and G. Laporte. The preemptive swapping problem on a tree. Technical report, Les Cahiers du GERAD, G-2005-69, 2006.

[3] S. Anily and R. Hassin. The swapping problem. *Networks (New York, NY)*, 22(4) :419–433, 1992.

[4] C. Archetti, R. Mansini, and M.G. Speranza. Complexity and Reducibility of the Skip Delivery Problem. *Transportation Science*, 39(2) :182–187, 2005.

[5] C. Archetti and M.G. Speranza. *The Vehicle Routing Problem : Latest Advances and New Challenges*, chapter The Split Delivery Vehicle Routing Problem : A Survey, pages 103–122. Springer US, 2008.

[6] C. Archetti, M.G. Speranza, and A. Hertz. A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. *Transportation Science*, 40(1) :64–73, 2006.

[7] C. Archetti, M.G. Speranza, and M.W.P. Savelsberg. An Optimization-Based Heuristic for the Split Delivery Vehicle Routing Problem. *Transportation Science*, 42(1) :22–31, 2008.

[8] N. Ascheuer, M. Fischetti, and M. Grötschel. A Polyhedral Study of the Asymmetric Traveling Salesman Problem with Time Windows. *Networks*, 36(1) :69–79, 2000.

[9] M.J. Atallah and S.R. Kosaraju. Efficient Solutions to Some Transportation Problems with Applications to Minimizing Robot Arm Travel. *SIAM Journal on Computing*, 17 :849–869, 1988.

[10] A. Atamtürk and D. Rajan. On capacitated network design cut-set polyhedra. *Mathematical Programming*, 92 :315–333, 2002.

[11] F. Barahona. Network Design Using Cut Inequalities. *SIAM J. on Optimization*, 6(3) :823–837, 1996.

[12] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price : Column generation for solving huge integer programs. *Operations Research*, 46(3) :316–329, 1998.

[13] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal of Computing*, 6 :126–140, 1994.

[14] J.M. Belenguer, M.C. Martinez, and E. Mota. A lower bound for the split delivery vehicle routing problem. *Operations Research*, 48(5) :801–810, 2000.

[15] R. Bent and P. Van Hentenryck. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33 :875–893, 2006.

[16] G. Berbeglia, J.F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems : a classification scheme and survey. *TOP*, 15(1) :1–31, 2007.

[17] L. Bianchi. Notes on Dynamic Vehicle Routing - The State of the Art -. Technical Report IDISIA-05-01, IDISIA, 2000.

[18] D. Bienstock, S. Chopra, O. Günlük, and C.Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81 :177–199, 1998.

[19] C. Bordenave, M. Gendreau, and G. Laporte. A Branch-and-Cut Algorithm for the Non-Preemptive Swapping Problem. *Soumis à Naval Research Logistics*, 2008.

[20] C. Bordenave, M. Gendreau, and G. Laporte. A Branch-and-Cut Algorithm for the Preemptive Swapping Problem. Technical Report CIRRELT-2008-23, 2008.

[21] C. Bordenave, M. Gendreau, and G. Laporte. Heuristics for the mixed swapping problem. Technical Report CIRRELT-2008-24, 2008.

[22] V.E. Campos and E. Mota. Heuristic procedures for the capacitated vehicle routing problem. *Computational Optimization and Applications*, 16(3) :265–277, 2000.

[23] S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, 1971. Association for Computing Machinery.

[24] J.F. Cordeau, G. Laporte, and S. Ropke. *The Vehicle Routing Problem : Latest Advances and New Challenges*, chapter Recent Models and Algorithms for One-to-One Pickup and Delivery Problems, pages 103–122. Springer US, 2008.

[25] C.E. Cortés, M Matamala, and C. Contardo. The Pickup and Delivery Problem with Transfers : Formulation and Solution Approaches. In *VII French - Latin American Congress on Applied Mathematics*, 2005.

[26] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[27] G.B. Dantzig and R.H. Ramser. The Truck Dispatching Problem. *Management Science*, 6 :80–91, 1959.

[28] M. Dror, G. Laporte, and P. Trudeau. Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50 :239–254, 1994.

[29] M. Dror and P. Trudeau. Savings by Split Delivery Routing. *Transportation Science*, 23(2) :141–145, 1989.

[30] M. Dror and P. Trudeau. Split Delivery Routing. *Naval Research Logistics*, 37 :383–402, 1990.

[31] Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54 :7–22, 1991.

[32] J. Edmonds. Covers and packings in a family of sets. *Bull. American Mathematical Society*, 68(5) :494–499, 1962.

[33] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, 69 :125 – 130, 1965.

[34] J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2) :248–264, 1972.

[35] H.A. Eiselt, M. Gendreau, and G. Laporte. Arc Routing Problems, Part I : the Chinese Postman Problem. *Operations Research*, 43(2) :231–242, 1995.

[36] M.T. Fiala Timlin. Precedence constrained routing and helicopter scheduling. *M. Sc. Thesis, Department of Combinatorics and Optimization University of Waterloo*, 1989.

[37] M.T. Fiala Timlin and W.R. Pulleyblank. Precedence constrained routing and helicopter scheduling : heuristic design. *Interfaces*, 22 :100–111, 1992.

[38] G.N. Frederickson and D.J. Guan. Preemptive Ensemble Motion Planning on a Tree. *SIAM Journal on Computing*, 21 :1130, 1992.

[39] G.N. Frederickson and D.J. Guan. Nonpreemptive Ensemble Motion Planning on a Tree. *Journal of Algorithms*, 15(1) :29–60, 1993.

[40] G.N. Frederickson, M.S. Hecht, and C.E. Kim. Approximation Algorithms for Some Routing Problems. *SIAM Journal on Computing*, 7 :178, 1978.

[41] V. Gabrel, A. Knippel, and M. Minoux. Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letter*, 25(1) :15–23, 1999.

[42] M.R. Garey and D.S. Johnson. Computers and Intractability : A Guide to the Theory of NP-Completeness. *W. H. Freeman and Company, New-York*, 1979.

[43] M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6) :1086 – 1094, 1992.

[44] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimisation. *Combinatorica 1*, pages 70–89, 1981.

[45] T. Grünert and H.J. Sebastian. Planning models for long-haul operations of postal and express shipment companies. *European Journal of Operational Research*, 122(2) :289–309, 2000.

[46] D.J. Guan. Routing a vehicle of capacity greater than one. *Discrete Applied Mathematics*, 81 :41–57, 1998.

[47] H. Hernández-Pérez and J.J. Salazar-González. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics*, 145(1) :126–139, 2004.

[48] H. Hernández-Pérez and J.J. Salazar-González. Heuristics for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem. *Transportation Science*, 38(2) :245–255, 2004.

[49] H. Hernández-Pérez and J.J. Salazar-González. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 196(3) :987–995, 2009.

[50] I. Ioachim, J. Desrosiers, Y. Dumas, M.M. Solomon, and D. Villeneuve. A Request Clustering Algorithm for Door-to-Door Handicapped Transportation. *Transportation Science*, 29(1) :63–78, 1995.

[51] M. Iri. On an Extension of the Maximum-flow Minimum-cut Theorem to Multicommodity Flows. *Journal of the Operations Research Society of Japan*, 13(3) :129–135, 1971.

[52] M. Jin, K. Liu, and B. Eksioglu. A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, 36 :265–270, 2008.

[53] B. Kalantari, A.V. Hill, and S.R. Arora. An algorithm for the traveling salesman problem with pickup and delivery customers. *European Journal of Operational Research*, 22 :377–386, 1985.

[54] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4) :373–395, December 1984.

[55] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Dokl.*, 20 :191–194, 1979.

[56] H.C. Lau and Z. Liang. Pickup and delivery with time windows : algorithms and test case generation. *International Journal on Artificial Intelligence Tools*, 11(3) :455–472, 2002.

[57] S. Lin. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, 44 :2245–2269, 1965.

[58] S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21(2) :498–516, 1973.

[59] J. Little, K. Murty, D. Sweeney, and C. Karel. An algorithm for the traveling salesman problem. *Operations Research*, 11(6) :972–989, 1963.

[60] I. Loiseau, A. Ceselli, N. Maculan, and M. Salani. Génération de colonnes en programmation linéaire en nombres entiers. In V. Th. Paschos, editor, *Optimisation combinatoire : Concepts fondamentaux*. chapitre 8. Hermès, Paris, 2005.

[61] R. Lougee-Heimer. The common optimization interface for operations research. *IBM Journal of Research and Development*, 47(1) :57–66, 2003.

[62] Q. Lu and M. Dessouky. An Exact Algorithm for the Multiple Vehicle Pickup and Delivery Problem. *Transportation Science*, 38(4) :503–514, 2004.

[63] M.E. Lübbecke and J. Desrosiers. Selected Topics in Column Generation. *Operations Research*, 53(6) :1007–1023, 2005.

[64] T.L. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60 :233–250, 1993.

[65] K. Menger. Zur allgemeinen kurventheorie. *Fundamanta Mathematicae*, 10 :96–115, 1927.

[66] S. Mitrović-Minić. Pickup and Delivery Problem with Time Windows : A Survey. Technical Report SFU CMPT TR 1998-12, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, 1998.

[67] S. Mitrović-Minić and G. Laporte. The pickup and delivery problem with time windows and transshipment. *Information Systems and Operation Research*, 44 :217–227, 2006.

[68] L. Moreno, M. Poggi de Aragão, and E. Uchoa. Improved Lower Bounds for the Split Delivery Vehicle Routing Problem. *Operations Research Letters*, À paraître.

[69] D. Naddef and G. Rinaldi. *The Vehicle Routing Problem*, chapter Branch-and-Cut algorithms for the Capacitated VRP, pages 53–84. P. Toth et D. Vigo, SIAM Monograph on Discrete Mathematics and Applications, Philadelphia, 2002.

[70] W.P. Nanry and J.W. Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B*, 34 :107–121, 2000.

[71] G.L. Nemhauser and L.A. Wolsey. A recursive procedure to generate all cuts for 0-1 mixed integer programs. *Math. Program.*, 46(3) :379–390, 1990.

[72] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley-Interscience, 1999.

[73] M. Nowak, E. Özlem, and C.C. White. Pickup and Delivery with Split Loads. *Transportation Science*, 42(1) :32–43, 2008.

[74] P. Oertel. *Routing with Reloads.* Doktorarbeit, Universität zu Köln, 2000.

[75] K. Onaga and O. Kakhuso. On feasibility conditions of multicommodity flows networks. *Transactions on circuit theory*, 4 :425–429, 1971.

[76] S.N. Parragh, K.F. Doerner, and R.F. Hartl. A survey on pickup and delivery problems : Part II : Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(2) :81–117, 2008.

[77] J. Plesnik. The NP-Completeness of the Hamiltonian Cycle Problem in Planar Digraphs with Degree Bound Two. *Information Processing Letters*, 8(4) :199–201, 1979.

[78] H.N. Psaraftis. A Dynamic Programming Solution for the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. *Transportation Science*, 14(2) :130–154, 1980.

[79] H.N. Psaraftis. An exact algorithm for the Single Vehicle Many-to-Many Dial-a-Ride Problem with Time Windows. *Transportation Science*, 17(3) :351–357, 1983.

[80] W.R. Pulleyblank. *Handbooks in Operations Research and Management Science*, chapter Polyhedral combinatorics, pages 371–446. 1989.

[81] M. Queyranne and A.S. Schulz. Polyhedral approaches to machine scheduling. Technical Report 408/1994, 1994.

[82] G. Reinelt. TSPLIB - A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4) :376–384, 1991.

[83] S. Ropke, J.F. Cordeau, and G. Laporte. Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows. *Networks*, 49(4) :258–272, 2007.

[84] S. Ropke and D. Pisinger. An Adaptative Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4) :455–472, 2006.

[85] K.M. Ruland and E.Y. Rodin. The pickup and delivery problem : Faces and branch-and-cut algorithm. *Computers and Mathematics with Applications*, 33 :1–13, 1997.

[86] M.W.P. Savelsberg. An efficient implementation of local search algorithms for constrained routing problems. *European Journal of Operational Research*, 47(1) :75–85, 1990.

[87] M.W.P. Savelsberg and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1) :17–29, 1995.

[88] M.W.P. Savelsberg and M. Sol. DRIVE : Dynamic Routing of Independant Vehicles. *Operations Research*, 46(4) :474–490, 1998.

[89] A. Schrijver. Theory of linear and integer programming. *Wiley Interscience, Chichester*, 1986.

[90] A. Schrijver. *Combinatorial Optimization : Polyhedra and Efficiency (Algorithms and Combinatorics)*. Springer book, 2004.

[91] P. Shaw. A new local search algorithm providing high quality solutions to vehicles routing problems. Technical report, Departement of Computer Science, University of Strathclyde, Scotland, 1997.

[92] J. Siek, L.Q. Lee, and A. Lumsdaine. Boost graph library. http ://www.boost.org/libs/graph/, June 2000.

[93] M. Sigurd, D. Pisinger, and M. Sig. Scheduling Transportation of Live Animals to Avoid the Spred of Diseases. *Transportation Science*, 38(2) :197–209, 2004.

[94] R. Sirdey and H.L.M. Kerivin. Polyhedral combinatorics of a resource-constrained ordering problem part I : on the partial linear ordering polytope. Technical report, Technical Report PE/BSC/INF/017912 V01, 2007.

[95] K.R. Smilowitz, A. Atamtürk, and C.F. Daganzo. Deferred Item and Vehicle Routing within Integrated Networks. *Transportation Research Part E : Logistics and Transportation Review*, 39(4) :305–323, 2003.

[96] L.J.J. Van der Bruggen, J.K. Lenstra, and P.C. Schuur. Variable-Depth Search for the Single-Vehicle Pickup and Delivery Problem with Time Windows. *Transportation Science*, 27(3) :298–311, 1993.

[97] F. Vanderbeck. *Decomposition and Column Generation for Integer Programming*. PhD thesis, Université catholique de Louvain, Louvain, Belgique, 1994.

[98] J. Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Applied Mathematics*, 61(1) :83–90, 1995.

[99] L.A. Wolsey. *Integer Programming*. John Wiley & Sons, Inc., New York, 1998.

[100] H. Xu, Z.L. Chen, S. Rajapogal, and S. Arunapuram. Solving a Practical Pickup and Delivery Problem. *Transportation Science*, 37(3) :347–364, 2003.