

THÈSE DE DOCTORAT EN INFORMATIQUE
-
CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS
LABORATOIRE CÉDRIC

**MODÉLISATION ET OPTIMISATION DU
DÉPLOIEMENT DES RÉSEAUX DE
TÉLÉCOMMUNICATIONS : APPLICATIONS AUX
RÉSEAUX D'ACCÈS CUIVRES ET OPTIQUES**

Mathieu TRAMPONT

19 octobre 2009

Table des matières

1	Génération de colonnes classique	16
1.1	Introduction	17
1.2	Modélisations appropriées	18
1.2.1	Décomposition de Dantzig-Wolfe	19
1.2.2	Modélisation sous forme de problème de couverture d'ensemble	20
1.3	Dualité et plans coupants	24
1.3.1	Dualité lagrangienne	24
1.3.2	Liens entre génération de colonnes et algorithmes de plans coupants	26
1.4	Problèmes de stabilisation	29
1.5	Méthodes de stabilisation	31
1.5.1	Méthode Boxstep	31
1.5.2	Méthode de stabilisation par point intérieur	34
1.6	Branch & Price	39
2	Génération de colonnes « centrale »	41
2.1	Algorithme d'Elzinga&Moore	41
2.2	L'algorithme de Betrò	45
2.3	Algorithme de Betrò et génération de colonnes	48
2.3.1	Choix du paramètre α et amélioration de la borne supérieure	49
2.3.2	Adaptation de la méthode de Boxstep à l'algorithme de Betrò	50
2.3.3	Adaptation de la méthode de stabilisation par point intérieur à l'algorithme de Betrò	52
2.4	Algorithme central de génération de colonnes stabilisée	55
2.5	Comparaison des méthodes classique et centrale	58
2.6	Conclusion	60

3	Optimisation des réseaux d'accès	61
3.1	Problèmes de conception de réseau d'accès	61
3.2	Problèmes de localisation discrète	64
3.2.1	Problème de localisation simple	64
3.2.2	Problème de localisation d'équipements avec capacités	65
3.3	Problèmes de localisation continue	66
3.3.1	Le problème de Weber	66
3.3.2	Le problème de Weber multi-source	67
3.3.3	Le problème de Weber avec réseau de transport alternatif	69
4	Problème de Weber multi-source	71
4.1	Estimation du nombre optimal d'équipements	72
4.1.1	Résolution du problème de localisation simple associé	72
4.1.2	Géométrie Stochastique	74
4.2	Heuristiques basées sur LOCALLOC	76
4.3	Une méthode de Recuit simulé	77
4.4	Méthodes de générations de colonnes	79
4.5	Comparaison des différentes méthodes	86
4.6	Conclusion	90
5	Application au RTC	91
5.1	Introduction	92
5.1.1	Architecture du Réseau Téléphonique Commuté	92
5.1.2	Problème de localisation de sous-répartiteurs	93
5.2	Modélisations	95
5.2.1	Modélisation à l'aide de la géométrie stochastique	95
5.2.2	Modélisation sous forme de programme mathématique	96
5.3	Comparaison des méthodes	98
5.3.1	Géométrie stochastique	99
5.3.2	Heuristique multi-phase	99
5.3.3	Heuristiques hybrides : géométrie stochastique et LOCALLOC	100
5.3.4	Recuit simulé	101
5.3.5	Obtention de bornes inférieures par génération de colonnes	101
5.3.6	Comparaison sur des instances générées aléatoirement	110
5.3.7	Expérimentations sur instances réelles	119
5.4	Conclusion	121

6	Application à la FTTH	122
6.1	Introduction	122
6.1.1	Le réseau d'accès FTTH	123
6.1.2	Problématiques et enjeux pour France Télécom	125
6.2	Localisation de coupleurs optiques	128
6.2.1	Problèmes proches	129
6.2.2	Modèle de base	130
6.2.3	Réduction du graphe sous-jacent	136
6.2.4	Dimensionnement des coupleurs	141
6.2.5	Dimensionnement de l'OLT	146
6.2.6	Architecture à trois niveaux de coupleurs	146
6.2.7	Création de nouveau génie civil et connexions point-à-point	146
6.2.8	Expérimentations numériques	152
6.2.9	Perspectives	155
6.3	Choix du câblage	156
6.3.1	Contrainte d'éligibilité	156
6.3.2	Arborescence de câblage	157
6.3.3	Première modélisation	162
6.3.4	Problème de la modularité des câbles	169
6.3.5	Contraintes sur le nombre de câble	170
6.3.6	Modélisation alternative des câbles	171
6.3.7	Reconstruction d'une solution opérationnelle	172
6.4	Conclusion et perspectives	174

Table des figures

1.1	Exemples d'hyperplans (ici des droites) correspondant à différentes valeurs du vecteur \mathbf{z} et représentation de la fonction de Lagrange . .	25
1.2	Les étapes de la génération de colonnes et leurs contreparties du côté du dual, dans le cas d'une décomposition de Dantzig-Wolfe	27
1.3	Exemple de ralentissement de la convergence des valeurs duales . . .	30
1.4	Problème de convergence lié à la dégénérescence du primal	30
1.5	Utilisation d'une méthode de boxstep avec une taille fixe d sur l'exemple de la figure 1.3	32
1.6	Illustration du choix d'un point intérieur pour le vecteur dual \mathbf{u}^k . .	35
2.1	Interprétation géométrique des trois inégalités de borne inférieure. À gauche l'inégalité 2.2, au centre 2.3, et à droite 2.4	43
2.2	Illustration du déroulement de l'algorithme de Betró lorsque $(\mathcal{R}\mathcal{B}etro_k)$ est faisable, pour $\alpha = 0,5$	47
2.3	Illustration du déroulement de l'algorithme de Betró lorsque $(\mathcal{R}\mathcal{B}etro_k)$ est infaisable, pour $\alpha = 0,5$	47
2.4	Fonctionnement de la méthode de génération de colonnes centrale implémentée	57
2.5	Évolution des bornes sur l'instance 14 avec l'initialisation la moins bonne	60
3.1	Une solution d'un problème de Weber multi-source	68
4.1	Exemples de réalisations de processus ponctuels homogène de Poisson pour différentes valeurs de λ	74
4.2	Adaptation du modèle de réseau hiérarchique à 3 niveaux pour revenir à un problème de Weber multi-source	75
4.3	Différence dans l'homogénéité de la distribution des clients selon l'échelle	76

4.4	Exemple de problème de Weber avec distances maximales	82
4.5	Problème de Weber avec distances maximales où la solution est d'installer l'équipement sur le client isolé 4	82
4.6	Exemple de cas où les décisions de branchement peuvent empêcher de trouver la solution optimale du sous-problème	87
4.7	Topologies des instances p654 et u1060	88
5.1	Partie Accès du RTC pour une zone locale	92
5.2	Exemple de répartition des éléments du RTC dans le plan	93
5.3	Structure des coûts du réseau d'accès	94
5.4	Représentation par des ellipses des ovales de Descartes induits par la propriété 5.1	106
5.5	Intersection de deux ovales de Descartes partageant le foyer N	107
5.6	Les 4 régions du plan auxquelles appartient le point d'intersection M et les groupes de clients correspondants.	109
5.7	Méthodologie des tests sur les instances artificielles	110
5.8	Évolution de l'écart avec l'heuristique multi-phase et du temps de calcul selon la taille des instances	118
5.9	Comparaison du nombre moyen de SR dans les solutions	118
5.10	Ajustement de la surface considérée pour le calcul des densités sur DAN120	
6.1	Les différentes architectures FTTx	124
6.2	Les éléments de réseau d'accès FTTH dans l'architecture actuelle . .	126
6.3	Exemple d'instance avec une solution admissible	129
6.4	Représentation du graphe G' dans le cas de l'exemple de la figure 6.3	131
6.5	Illustration d'une inégalité valide obtenue par rotation	135
6.6	Exemple de solution de la relaxation continue de (\mathcal{LCOdim}) avec des variables fractionnaires, avec des coupleurs 1 :8 et des taux de dimensionnement de 16% et 25%	145
6.7	Comparaison des déploiements selon le nombre de clients	145
6.8	Architecture GPON à trois niveaux de coupleurs optiques	147
6.9	Effet des différentes réductions du graphe sur l'instance AIG	154
6.10	Topologie de la solution heuristique de déploiement pour AIG	155
6.11	Exemple de topologie du génie civil existant	157
6.12	Exemple de solution de déploiement	158
6.13	Arborescence pour le déploiement de la figure 6.12	158

6.14 Exemple de situation où il faut décider sur quels câbles les épissures sont pratiquées	162
6.15 Illustration des contraintes (6.44) et (6.45) d'une part, et (6.46) d'autre part, avec des coupleurs de capacité 4.	165
6.16 Cas où le passage des fibres dans l'arborescence utilise trop de coupleurs de niveau 2	166
6.17 Illustration du piquage	166
6.18 Illustration d'une solution irréalisable à cause de la modularité des câbles (ici 12)	169
6.19 Exemple de cas où l'indivisibilité des modules amène à devoir en installer deux au lieu d'un seul	169
6.20 Exemple de graphe du sous-problème	173

Liste des tableaux

2.1	Temps de calcul des deux générations de colonnes avec différentes méthodes de stabilisation	58
2.2	Résultats numériques sur deux séries de dix instances	59
2.3	Temps CPU consommé avec une initialisation moins bonne	59
4.1	Résultats des différentes méthodes sur les instances p654 et u1060	89
5.1	Résultats de l'heuristique multi-phase sur des instances de taille moyenne	112
5.2	Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 1250 clients	113
5.3	Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 1680 clients	114
5.4	Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 2250 clients	115
5.5	Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 2950 clients	116
5.6	Résultats sur deux instances issues de données réelles	120
6.1	Présentation des principales caractéristiques des instances GLO et BDV	153

Résumé

Ce manuscrit présente des travaux de recherche effectués pour Orange Labs sur l'optimisation des réseaux d'accès télécoms. Il s'agit de la partie du réseau qui agrège le trafic des clients se trouvant dans une même zone géographique, avant de le transmettre vers le coeur du réseau. C'est un réseau hiérarchique avec à son sommet un concentrateur central et à sa base les clients, et qui présentent un ou plusieurs niveaux de concentrateur intermédiaires.

Lors du déploiement de nouveaux réseaux, la plus grande part du coût est celle du réseau d'accès, notamment du fait des travaux de génie civil nécessaires. L'optimisation de ces déploiements est donc un enjeu financier majeur. Malheureusement les problèmes d'optimisation rencontrés dans les réseaux d'accès sont difficiles, et la taille des instances posent souvent problème. Nous nous intéressons ici à deux applications distinctes : l'optimisation de l'accès du réseau téléphonique commuté sur un territoire vierge, et l'optimisation du déploiement d'un réseau d'accès en fibre optique en présence d'une infrastructure existante.

Pour la première application, nous proposons d'améliorer les méthodes de résolution existantes et proposons de nouvelles méthodes afin d'avoir des méthodes fournissant des solutions de qualité pour différents types d'instance. Ces travaux nous ont amené à développer une nouvelle méthode de génération de colonnes « centrale » (méthode d'optimisation pour problème de grande taille basée sur une décomposition du problème) dont la portée dépasse notre application. Nous avons également adapté les différentes méthodes utilisées à un problème de localisation continue classique nommé problème de Weber multi-source.

Dans le cas du déploiement du réseau d'accès optique, les technologies et les architectures utilisées font que les modèles existant dans la littérature ne conviennent pas. Le problème consiste à raccorder les clients au concentrateur central à travers deux ou trois niveaux de coupleurs optiques, en utilisant différentes taille de câbles optiques. Le but de l'optimisation est de minimiser le coût de déploiement global en plaçant les coupleurs, en décidant de la topologie du réseau, et en choisissant les câbles utilisés. Tout cela se fait en tenant compte d'éventuelles conduites libres existantes qui permettent d'économiser sur le génie civil. Nous séparons ce problème d'optimisation en

localisation des coupleurs d'une part, et en choix du câblage d'autre part. Nous proposons des modèles pour chacune des problématiques et pour les différents aspects opérationnels qu'il est possible de prendre en compte. Quelques tests sont présentés montrant qu'il est possible d'obtenir des solutions de qualité sur des instances réelles pour le problème de localisation de coupleurs.

Introduction

Aujourd'hui, les opérateurs de télécommunications, pour conquérir de nouveaux marchés, pour fournir de nouveaux services, ou encore pour faire face à l'augmentation du trafic, déploient de nouveaux réseaux ou redéploient et adaptent les réseaux existants. Les énormes investissements nécessaires poussent ces opérateurs à optimiser autant que possible ces déploiements. Ils font alors face à des problèmes complexes. Ces problèmes sont différents selon la partie du réseau qui est considérée. Une première décomposition des réseaux en deux est donc réalisée :

Réseau coeur : Ce réseau achemine les différents trafics et contient les équipements associés aux différents services. Il s'occupe aussi de l'interconnexion des différents réseaux (réseaux d'opérateurs tiers, réseaux de différentes technologies, réseaux d'accès). Les problématiques d'optimisation associées sont complexes : routage, dimensionnement, survivabilité, multiplexage, collecte,... mais les réseaux sont de taille moyenne (entre quelques dizaines et quelques centaines de noeuds).

Réseau d'accès : Ce réseau relie les abonnés au réseau coeur. Il s'agit d'un ensemble de réseaux hiérarchiques qui agrègent le trafic des abonnés vers des concentrateurs faisant la transition vers le coeur du réseau. Chacun de ces concentrateurs est la racine d'un arbre où le trafic des abonnés est agrégé au cours des passages successifs par les différents niveaux d'équipements intermédiaires. Un grand nombre d'abonnés est ainsi raccordé à un même concentrateur. Par exemple, dans le cas du réseau téléphonique commuté, la dizaine de milliers d'abonnés d'une petite ville est reliée à son concentrateur par le biais de sous-répartiteurs répartis dans la ville.

Nous nous intéressons dans ce manuscrit à l'optimisation du déploiement du réseau d'accès. Il s'agit de localiser les équipements intermédiaires (et éventuellement de les dimensionner), et de raccorder les différents éléments.

L'optimisation de ce « dernier kilomètre » (« *last mile* ») est d'autant plus importante que sa part dans le coût total du réseau est la plus importante. Son coût énorme est dû notamment aux travaux de génie civil nécessaires (notamment pour les réseaux fixes).

La principale difficulté de l'optimisation du réseau d'accès, est la taille des données à prendre en compte. En effet, même en ne considérant qu'un seul équipement central, c'est le raccordement de plusieurs milliers de clients qui doit être optimisé. À cette difficulté de taille viennent s'ajouter les contraintes techniques liées aux technologies et la prise en compte des infrastructures existantes, avec d'éventuelles trajectoires de migration vers le nouveau réseau.

La complexité du déploiement rend la solution optimale hors d'atteinte. Dès lors, il est nécessaire d'établir un compromis entre la qualité des solutions et les efforts nécessaires pour les obtenir.

Pour atteindre ce compromis, les opérateurs cherchent à élaborer des règles d'ingénierie qui se focalisent sur un aspect particulier du déploiement (localisation des équipements d'un niveau hiérarchique donné, dimensionnement, type de médium utilisé,...) en effectuant des hypothèses sur les autres. Le but est d'atteindre une solution de qualité en mettant en oeuvre l'ensemble de ces règles.

Il est nécessaire de valider ces règles et il est possible de les améliorer. Classiquement, cette validation se fait en utilisant des modèles simplifiés. Par exemple, il peut s'agir de méthodes statistiques qui permettent de représenter les données à l'aide de peu de paramètres, ou bien d'optimiser séparément chaque niveau du réseau hiérarchique à l'aide de méthodes de recherche opérationnelle. La complexité des problèmes fait que le potentiel d'amélioration est élevé.

Les déploiements se font dans des contextes différents selon qu'il s'agit d'un territoire vierge, ou d'un territoire possédant des infrastructures existantes. Lorsque le territoire est vierge, seuls les emplacements des abonnés et de l'équipement central sont connus. Les liens entre les éléments du réseau se font souvent à vol d'oiseau pour simplifier. Le déploiement d'un réseau sur un territoire possédant des infrastructures existantes peut signifier deux choses : l'extension d'un réseau existant pour répondre à une augmentation de trafic ou une évolution des abonnés, ou le déploiement d'un nouveau réseau en utilisant le génie civil (les conduites, les poteaux, les chambres de béton, ...) laissé disponible par un réseau existant.

Cette thèse se concentre sur deux applications. Dans la première, il s'agit de déployer le RTC¹ sur un territoire vierge. La deuxième application, et celui d'un réseau d'accès optique FTTH² (la fibre optique va de l'équipement central jusqu'au logements) en réutilisant le génie civil existant.

Dans le RTC, le réseau hiérarchique est composé de 3 niveaux : Le NRA³ au plus haut ; les sous-répartiteurs ; les abonnés au plus bas. L'objectif est de décider du nombre de sous-répartiteurs et de les localiser sur le territoire. Ensuite, chaque abonné est raccordé à un sous-répartiteur, et enfin les sous-répartiteurs sont raccordés au NRA.

1. Réseau Téléphonique Commuté

2. Fiber To The Home

3. Noeud de Raccordement des Abonnés

Le territoire étant vierge, il n'y a pas d'emplacements potentiels prédéfinis pour les sous-répartiteurs. Le coût du déploiement regroupe les coûts d'installation des sous-répartiteurs, les coûts des paires de cuivre, et le coût des tranchées et des poteaux pour faire passer ces paires de cuivre. L'optimisation a pour but la minimisation du coût global. Ce problème présente des ressemblances avec un problème de localisation connu : le problème de Weber multi-source, où il n'y a que deux niveaux.

Une manière simple d'analyser la répartition des éléments du réseau sur le territoire, est d'utiliser des outils statistiques simples pour les représenter. Ces modèles disposent de différents paramètres pour chaque niveau d'équipements (par exemple, le nombre moyen d'équipements par unité de surface). Ces paramètres peuvent être affinés en fonction des situations opérationnelles (par exemple différentes densités de clients pour Paris ou pour une zone rurale).

Ces modèles sont utiles pour capturer certaines caractéristiques géographiques du déploiement ou obtenir des jeux d'essai rapidement. Mais ils ne représentent que partiellement la réalité et ne permettent pas d'optimiser le déploiement en tenant compte des emplacements précis des abonnés d'une zone donnée.

Notre objectif est d'améliorer l'optimisation en prenant en compte la répartition réelle des abonnés de chaque instance. Nous proposons d'établir un panel de méthodes appropriées aux zones de différentes tailles en proposant des méthodes de recherche opérationnelle. Nous nous assurons également de la qualité des solutions retournées par les différentes méthodes.

Pour cela, nous avons adapté des méthodes utilisées pour le problème de Weber multi-source. Nous avons également développé deux nouvelles méthodes : une heuristique s'appuyant sur les outils statistiques évoqués précédemment, et une méthode de génération de colonnes (technique de recherche opérationnelle s'appuyant sur une décomposition du problème). Cette dernière sépare le problème en deux. Le problème maître choisit parmi une partie des regroupements possibles de clients ceux permettant de minimiser le coût. Le sous-problème est chargé de trouver les regroupements de clients intéressants à ajouter au problème maître.

La génération de colonnes est souvent utilisée pour l'optimisation de problèmes de grandes tailles. Elle est toutefois souvent confrontée à des problèmes de convergence auxquels nous avons consacré une attention toute particulière. Cela nous a amené à développer une nouvelle méthode de génération de colonnes basée sur un algorithme de plans coupants central, qui de ce fait diffère de la génération de colonnes classique. Il est également à noter que la portée de la nouvelle génération de colonnes centrale dépasse les problèmes d'optimisation dans les réseaux.

La deuxième application porte sur le déploiement d'un réseau d'accès optique jusqu'à l'abonné (*Fiber To The Home*) en présence d'une infrastructure existante.

Devant l'augmentation du trafic internet et afin de proposer de nouveaux services, les opérateurs installent de nouveaux réseaux d'accès où la fibre optique se rapproche de

plus en plus des abonnés. Pour réaliser des économies sur les travaux de génie civil, les opérateurs cherchent à utiliser les infrastructures existantes. Ces infrastructures sont celles d'autres réseaux de télécommunications où il reste de la place pour faire passer les nouveaux câbles. Dans certaines villes, certaines infrastructures urbaines, comme les égouts, peuvent également être utilisées.

Dans les zones urbaines, les travaux de génie civil coûtent extrêmement cher. De plus, de nombreuses conduites sont déjà présentes. Par conséquent, nous voulons nous contenter de l'espace disponible dans ces conduites.

L'architecture choisie pour notre réseau d'accès FTTH comprend deux niveaux de coupleurs optiques entre l'équipement central et les abonnés. Les coupleurs optiques sont des équipements répartissant le débit d'une fibre en entrée sur plusieurs fibres en sortie. Le déploiement consiste donc à localiser les coupleurs optiques dans les chambres de béton, à raccorder chaque abonné à un coupleur de niveau 2, lui-même raccordé à un coupleur de niveau 1, lui-même raccordé à l'équipement central. Les modèles existants dans la littérature (abordés dans le chapitre 3) ne conviennent pas à ces spécificités.

Notre proposons donc des modélisations et des méthodes de résolution permettant d'étudier, de vérifier, et éventuellement d'améliorer, les règles d'ingénierie actuellement mises en place.

Ces règles concernent différents aspects du déploiements. Certaines concernent les coupleurs, plus précisément leur localisation et leur dimensionnement. D'autres concernent les câbles et précisent leur type et la façon de les raccorder entre eux. D'autres règles visent à respecter des contraintes techniques, comme par exemple la distance maximale entre l'équipement central et un abonné.

Prendre l'ensemble de ces aspects en compte dans un même modèle est trop complexe. Nous avons donc séparé notre problème en deux : localisation de coupleurs optiques d'une part, et choix du câblage d'autre part. Le premier problème fait, entre autres, l'hypothèse que les fibres peuvent être installées une par une. Le deuxième problème corrige cette simplification en choisissant les câbles qui seront utilisés et en tenant compte des contraintes spécifiques sur ces câbles.

Ce manuscrit s'articule en six chapitres. Les deux premiers chapitres portent sur la génération de colonnes. Le premier présente des rappels nécessaires sur cette méthode et sa stabilisation. Le chapitre 2 présente la nouvelle génération de colonnes centrale en détails. L'algorithme de plans coupants central utilisé est introduit, ainsi que son adaptation à la génération de colonnes. Nous présentons également l'adaptation à cette nouvelle méthode des techniques de stabilisation présentées dans le chapitre 1. Cette nouvelle méthode est ensuite comparée à la génération de colonnes classique pour montrer son efficacité et sa stabilité.

Le chapitre 3 passe en revue les problèmes d'optimisation de réseaux d'accès dans la littérature et présente certains problèmes d'optimisation classiques liés à nos appli-

cations.

Le chapitre 4 est consacré à la résolution du problème de Weber multi-source, dans sa variante où le nombre d'équipements à placer est inconnu. Ce problème de localisation-allocation continue classique est difficile. Il présente des similarités avec l'application sur le RTC et les méthodes utilisées pour l'un peuvent être adaptées à l'autre. Nous présentons dans ce chapitre certaines heuristiques existantes, et en proposons une nouvelle, ainsi qu'une méthode de génération de colonnes permettant d'obtenir une borne inférieure. Des comparaisons de ces différentes méthodes sont alors présentées. La borne inférieure obtenue par génération de colonnes permet de montrer leur efficacité.

Les chapitres 5 est consacré au déploiement du sur RTC un territoire vierge. Le problème correspondant y est décrit, puis modélisé. Deux modélisations sont présentées : une modélisation existante basée sur des outils statistiques pour représenter les différents niveaux d'équipements, et une modélisation sous forme d'un programme mathématique. Différentes méthodes de résolution empruntant aux deux modélisations sont présentées. Ces méthodes sont comparées sur des instances artificielles de différentes tailles, ainsi que sur deux instances issues de données réelles. Nous obtenons ainsi différents compromis entre temps de calcul et qualité des solutions adaptés aux différentes instances.

Le chapitre 6 traite de l'optimisation du déploiement d'un réseau d'accès FTTH en tenant compte des infrastructures existantes. Après la description de ce problème complexe, nous présentons essentiellement des modélisations mathématiques pour les deux aspects de ce déploiement : le problème de localisation de coupleurs optiques, et le problème de choix de câblage.

Pour le premier problème, les spécificités de l'architecture de réseau FTTH choisie rendent les modèles existants inappropriés. Nous proposons donc des modélisations mathématiques pour les différents aspects opérationnels du problème. Ces modèles sont testés sur deux instances réelles. La qualité des solutions obtenue varient selon les instances et des efforts restent à faire pour améliorer la résolution du problème.

Pour le problème de choix de câblage, nous présentons des modèles prenant en compte les différentes contraintes opérationnelles. Ces modèles se basent sur une solution du problème de localisation de coupleurs et n'ont pas encore été testés.

Chapitre 1

Rappels sur la génération de colonnes

Une des principales difficultés rencontrées lors de l'utilisation de méthodes d'optimisation sur des instances réelles de problèmes de conception de réseau est la taille des données. Cette taille rend souvent les méthodes de résolution exactes inabordables et peut même rendre très difficile et très longue l'obtention d'une bonne solution à l'aide d'heuristiques.

Les méthodes de générations de colonnes peuvent être un moyen pour développer des méthodes de résolution exactes pour des problèmes de grande taille, ou au moins pour obtenir de bonnes bornes permettant de juger de la qualité d'autres méthodes. C'est pour ce second objectif que nous avons utilisé la génération de colonnes sur le problème du chapitre 5. La très grande taille des instances et les problèmes de stabilisation rencontrés par la génération de colonnes nous ont amené à développer une nouvelle méthode de génération de colonnes « centrale » présentée au chapitre 2, et dont la portée dépasse le seul problème du chapitre 5. Pour bien saisir les différences avec les méthodes de génération de colonnes classiques, nous présentons dans ce chapitre des rappels sur la génération de colonnes, ses liens avec les méthodes de plans coupants, et sur les méthodes de stabilisation qui seront utilisées au chapitre 2.

Dans ce chapitre, nous introduisons d'abord les principes de la génération de colonnes pour résoudre un programme linéaire. Nous présentons ensuite des façons d'obtenir une modélisation appropriée à l'utilisation de la génération de colonnes. Nous parlerons notamment de la décomposition de Dantzig-Wolfe et de formulations sous forme de problème de couverture d'ensembles. Ensuite nous présentons la génération de colonnes vue du dual et son lien avec les algorithmes de plans coupants. Ceci nous permet ensuite de parler des problèmes de stabilisation rencontrés en génération de colonnes et des méthodes pour s'y attaquer. Nous détaillerons notamment

deux méthodes de stabilisation que nous avons utilisées : la méthode boxstep et la méthode de stabilisation par point intérieur. Nous concluons par une brève introduction aux procédures de branch & price qui sont nécessaires pour résoudre des programmes linéaires en nombres entiers avec la génération de colonnes.

1.1 Introduction

Le principe de la génération de colonnes repose sur le fait que toutes les variables d'un programme linéaire ne sont pas forcément utiles pour atteindre la solution optimale. L'objectif est alors de résoudre un problème réduit en prenant en compte un ensemble limité de variables. Le problème initial est appelé problème maître, et le problème avec un nombre réduit de variables est appelé problème maître restreint. Le problème restreint est plus facile à résoudre, mais si l'ensemble de ses variables ne contient pas celles utilisées par la solution optimale du problème maître, nous n'obtiendrons pas cette dernière. Pour atteindre la solution optimale du problème maître, il faut rajouter au problème restreint des variables pouvant être utilisées pour améliorer la solution.

Le problème consistant à trouver la meilleure variable à rajouter dans le problème maître restreint est appelé sous-problème associé au problème maître. Ce sous-problème est aussi appelé oracle dans la littérature. Son but est de trouver la variable (ou colonne) de coût réduit minimum (*i.e.* la plus prometteuse pour améliorer la solution). Le coût réduit des variables est calculé à l'aide des variables duales obtenues après résolution du problème restreint. Le point du dual ainsi utilisé dans le sous-problème est appelé point de séparation. Le plus souvent, il s'agit d'une solution optimale du dual du problème restreint.

Prenons l'exemple du programme linéaire continu (\mathcal{P}) suivant :

$$(\mathcal{P}) \left\{ \begin{array}{l} \min_{\mathbf{z}} \quad \sum_{i \in S} c_i \cdot \mathbf{z}_i \\ \text{s.c.} \\ \sum_{i \in S} a_{ij} \cdot \mathbf{z}_i \geq b_j \quad \forall j = 1 \dots n \\ \mathbf{z}_i \geq 0 \quad \forall i \in S \end{array} \right. \quad (1.1)$$

Supposons que le nombre $|S|$ de variables soit trop grand pour que (\mathcal{P}) puisse être résolu en temps raisonnable par un solveur de programmation linéaire classique, et que nous voulions le résoudre par génération de colonnes.

Nous allons donc résoudre un problème maître restreint avec un ensemble restreint $I_k \subset S$ de variables. Le choix de l'ensemble des colonnes initiales I_0 est un point

important. Il faut que le problème restreint ne soit pas infaisable. Il est possible d'utiliser des colonnes simples (aisément constructibles), des colonnes aléatoires, ou encore celles issues d'une solution réalisable obtenue à l'aide d'une heuristique. Si l'on ne dispose pas de solution réalisable, pour éviter l'infaisabilité, on peut utiliser des colonnes artificielles. On supposera ici que nous n'en avons pas besoin.

Notre problème restreint se présente donc sous la forme suivante :

$$(\mathcal{RP}_k) \begin{cases} \min_{\mathbf{z}} & \sum_{i \in I_k} (c_i \mathbf{z}_i) \\ \text{s.c.} & \\ & \sum_{i \in I_k} a_{ij} \mathbf{z}_i \geq b_j \quad \forall j = 1 \dots n \\ & \mathbf{z}_i \geq 0 \quad \forall i \in I_k \end{cases} \quad (1.1)$$

Le problème (\mathcal{RP}_k) est de taille réduite et sera résolu plus facilement par un solveur. Cette résolution nous fournira les valeurs optimales des variables duales u_j associées aux contraintes (1.1). Ces valeurs sont passées au sous-problème qui nous permet de trouver la ou les colonnes à rajouter dans l'ensemble I_k .

Pour savoir si une colonne permettrait de diminuer la valeur de l'objectif (et donc de l'améliorer), il nous faut calculer son coût réduit. Prenons par exemple la colonne z_i du problème maître (\mathcal{P}) , son coût réduit vaut :

$$\bar{c}_i = c_i - \sum_{j=1}^n a_{ij} u_j \quad (1.2)$$

(\mathcal{P}) étant un problème de minimisation, le sous-problème cherchera à minimiser ce coût réduit. Si le coût réduit minimum est positif, alors aucune colonne ne peut être ajoutée au problème restreint pour améliorer l'objectif. La solution optimale du problème maître restreint est donc une solution optimale du problème maître. Sinon, on rajoute une ou des colonnes parmi celles ayant un coût réduit négatif (mettant à jour I_k), et on résout le nouveau problème restreint (\mathcal{RP}_{k+1}) .

Pour résoudre des programmes linéaires en variables entières, la génération de colonnes doit être intégrée à un algorithme d'énumération de type Branch&Bound qui est alors appelé Branch&Price.

1.2 Modélisations appropriées pour la génération de colonnes

Certains problèmes se modélisent assez naturellement de façon appropriée pour utiliser la génération de colonnes, par exemple sous forme de problème de couverture

d'ensemble. Mais souvent il est nécessaire de faire appel à des techniques de reformulation particulières comme la décomposition de Dantzig-Wolfe (voir [26, 13] pour plus de détails).

1.2.1 Décomposition de Dantzig-Wolfe

Dans un programme linéaire en nombres entiers \mathcal{Q} , nous cherchons une solution z qui appartienne à l'ensemble $\mathcal{X} \subset \mathbb{N}^n$. Le principe est alors d'exprimer les éléments de \mathcal{X} :

- soit comme une combinaison convexe des points extrêmes de l'enveloppe convexe de \mathcal{X} (notée $\text{Conv}(\mathcal{X})$) plus une combinaison positive de rayons extrêmes de $\text{Conv}(\mathcal{X})$ (approche de convexification),
- soit comme un point de \mathcal{X} et une combinaison positive de rayons extrêmes entiers de $\text{Conv}(\mathcal{X})$ (approche de discrétisation).

Prenons l'exemple du problème suivant :

$$(\mathcal{Q}) \left\{ \begin{array}{l} \min_{\mathbf{x}} \quad \sum_{i=1}^n c_i \mathbf{x}_i \\ \text{s.c.} \\ A \cdot \mathbf{x} \geq \mathbf{b} \quad (1.3) \\ D \cdot \mathbf{x} \geq \mathbf{e} \quad (1.4) \\ \mathbf{x} \in \mathbb{N}^n \end{array} \right. \quad (1.3)$$

On peut choisir ici $\mathcal{X} = \{\mathbf{x} \in \mathbb{N}^n \mid D \cdot \mathbf{x} \geq \mathbf{e}\}$.

Commençons par l'approche de convexification. Notons maintenant $\{\mathbf{x}^p\}_{p \in P}$ l'ensemble des points extrêmes de $\text{Conv}(\mathcal{X})$, et $\{\mathbf{x}^r\}_{r \in R}$ l'ensemble de ses rayons extrêmes. On peut écrire $\mathbf{x} = \sum_{p \in P} \lambda_p \mathbf{x}^p + \sum_{r \in R} \mu_r \mathbf{x}^r$, $\sum_{p \in P} \lambda_p = 1$ où λ_p et μ_r sont des réels positifs. En notant $\mathbf{c}^k = \mathbf{c} \cdot \mathbf{x}^k$ et en intégrant tout cela dans le modèle précé-

dent on obtient :

$$(\mathcal{DW} - \mathcal{Q}) \left\{ \begin{array}{l} \min_{\lambda, \mu} \sum_{p \in P} \lambda_p c^p + \sum_{r \in R} \mu_r c^r \\ \text{s.c.} \\ \sum_{p \in P} \lambda_p A x^p + \sum_{r \in R} \mu_r A x^r \geq \mathbf{b} \quad (1.3') \\ \sum_{p \in P} \lambda_p = 1 \quad (1.5) \\ \sum_{p \in P} \lambda_p x^p + \sum_{r \in R} \mu_r x^r \in \mathbb{N}^n \\ \lambda_p, \mu_r \in \mathbb{R}^+ \end{array} \right.$$

Ce modèle est une formulation extensive du problème. La contrainte (1.3') correspond à la contrainte (1.3) et la contrainte (1.5) est appelée contrainte de convexité. Souvent les ensembles $\{x^p\}_{p \in P}$ et $\{x^r\}_{r \in R}$ sont de taille exponentielle par rapport à n . Le modèle sera donc résolu par génération de colonnes et les difficultés liées aux contraintes (1.4) se retrouveront déplacées dans le sous-problème.

L'approche de discrétisation aboutit à un modèle assez similaire. Notons $\{x^p\}_{p \in P'}$ un ensemble fini de points entiers de \mathcal{X} et $\{x^r\}_{r \in R'}$ l'ensemble des rayons extrêmes entiers de $\text{Conv}(\mathcal{X})$, on peut écrire :

$$\mathbf{x} = \sum_{p \in P'} \lambda_p x^p + \sum_{r \in R'} \mu_r x^r, \quad \sum_{p \in P'} \lambda_p = 1$$

où λ_p et μ_r sont des entiers positifs. On obtient alors le modèle suivant :

$$(\mathcal{DW} - \mathcal{Q}') \left\{ \begin{array}{l} \min_{\lambda, \mu} \sum_{p \in P} \lambda_p c^p + \sum_{r \in R} \mu_r c^r \\ \text{s.c.} \\ \sum_{p \in P} \lambda_p A x^p + \sum_{r \in R} \mu_r A x^r \geq \mathbf{b} \quad (1.3') \\ \sum_{p \in P} \lambda_p = 1 \quad (1.5) \\ \lambda_p, \mu_r \in \mathbb{N} \end{array} \right.$$

1.2.2 Modélisation sous forme de problème de couverture d'ensemble

Une application particulière de la décomposition de Dantzig-Wolfe est la reformulation sous forme d'un problème de couverture d'ensemble. Nous l'illustrerons ici sur le

problème de localisation simple. Il s'agit de trouver les emplacements d'équipements et l'affectation de clients à ces derniers de façon à minimiser la somme des coûts d'installation et d'affectation. Ce problème est modélisé de façon classique par le programme linéaire en nombres entiers suivant :

$$(LS) \left\{ \begin{array}{l} \min_{\mathbf{x}, \mathbf{y}} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} \mathbf{x}_{ij} + \sum_{i=1}^m f_i \mathbf{y}_i \\ \text{s.c.} \\ \mathbf{x}_{ij} \leq \mathbf{y}_i \quad \forall i = 1 \dots m, \forall j = 1 \dots n \quad (1.6) \\ \sum_{i=1}^m \mathbf{x}_{ij} \geq 1 \quad \forall j = 1 \dots n \quad (1.7) \\ \mathbf{x}_{ij}, \mathbf{y}_i \in \{0; 1\} \end{array} \right.$$

où :

- m est le nombre d'emplacements potentiels pour les équipements ;
- n est le nombre de clients ;
- \mathbf{y}_i vaut 1 lorsqu'un équipement est placé sur le site i , 0 sinon ;
- \mathbf{x}_{ij} vaut 1 lorsque le client j est affecté à l'équipement installé sur le site i , 0 sinon ;
- c_{ij} est le coût d'affectation du client j au site i ;
- f_i est le coût d'installation d'un équipement sur le site i ;
- Les contraintes (1.6) empêchent un client d'être rattaché à un site sans équipement ;
- Les contraintes (1.7) s'assurent que chaque client est affecté à au moins un site.

Au lieu de considérer chaque client individuellement pour leur affectation, il est possible de les considérer par groupe de clients affectés à un même site. Notons E l'ensemble des clients, et $\mathcal{P}(E)$ l'ensemble des parties de E . Le nombre de groupes de clients possibles est égal à $|\mathcal{P}(E)| - 1 = 2^n - 1$. Pour chacun de ces groupes, m affectations sont possibles. Le problème peut donc se formuler sous la forme du problème de couverture d'ensemble suivant :

$$(LS_{set}) \left\{ \begin{array}{l} \min_{\mathbf{z}} \quad \sum_{k=1}^{m(2^n-1)} c^k \mathbf{z}_k \\ \text{s.c.} \\ \sum_{k=1}^{m(2^n-1)} x_j^k \mathbf{z}_k \geq 1 \quad \forall j = 1 \dots n \quad (1.8) \\ \mathbf{z}_k \in \{0; 1\} \end{array} \right.$$

- \mathbf{z}_k vaut 1 si le groupe k est utilisé, 0 sinon ;

- x_j^k vaut 1 si le groupe k contient le client j , 0 sinon ;
- $c^k = \sum_{i=1}^m y_i^k \left(f_i + \sum_{j=1}^n x_j^k c_{ij} \right)$, où y_i^k vaut 1 si le groupe k utilise un équipement placé sur le site i . c^k correspond donc au coût d'utilisation du groupe k .

Ce modèle présentant un nombre exponentiel de variables peut être résolu par génération de colonnes et branch-and-price.

Le modèle (LS_{set}) peut être obtenu à partir du modèle (LS) en appliquant l'approche de discrétisation de la décomposition de Dantzig-Wolfe m fois sur les ensembles suivant : $Z_i = \left\{ (\mathbf{x}_i, y_i) \in \{0, 1\}^{n+1} \mid x_{ij} \leq y_i, \forall j = 1 \dots n \right\}$, pour tous les sites i . L'ensemble des points entiers de Z_i représente les différents groupes de clients pouvant être affectés au site i . Appliquons Dantzig-Wolfe à l'ensemble Z_1 . Nous obtenons :

$$(\mathcal{DW} - LS_1) \left\{ \begin{array}{l} \min_{\mathbf{z}, \mathbf{x}, \mathbf{y}} \quad \sum_{k=1}^{2^n+1} c^{1k} \mathbf{z}_{1k} + \sum_{i=2}^m \sum_{j=1}^n c_{ij} \mathbf{x}_{ij} + \sum_{i=2}^m f_i \mathbf{y}_i \\ \text{s.c.} \\ \mathbf{x}_{ij} \leq \mathbf{y}_i \quad \forall i = 2 \dots m, \forall j = 1 \dots n \quad (1.6') \\ \sum_{k=1}^{2^n+1} x_{1j}^k \mathbf{z}_{1k} + \sum_{i=2}^m \mathbf{x}_{ij} \geq 1 \quad \forall j = 1 \dots n \quad (1.7') \\ \sum_{k=1}^{2^n+1} \mathbf{z}_{1k} = 1 \quad (1.9) \\ \mathbf{x}_{ij}, \mathbf{y}_i, \mathbf{z}_{1k} \in \{0; 1\} \end{array} \right.$$

- \mathbf{x}_1^k est le vecteur caractéristique du k -ième élément de l'ensemble Z_1 , x_{1j}^k vaut 1 si ce k -ième groupe contient le client j , 0 sinon ;
- $c^{1k} = y_1^k f_1 + \sum_{j=1}^n x_{1j}^k c_{1j}$ correspond au coût d'utilisation du groupe associé à \mathbf{z}_{1k} .

En continuant à appliquer Dantzig-Wolfe sur tous les autres ensembles Z_i , nous obtenons alors le modèle suivant :

$$(DW - LS) \left\{ \begin{array}{l} \min_{\mathbf{z}} \sum_{i=1}^m \sum_{k=1}^{2^n+1} c^{ik} \mathbf{z}_{ik} \\ \text{s.c.} \\ \sum_{k=1}^{2^n+1} \sum_{i=1}^m x_{ij}^k \mathbf{z}_{ik} \geq 1 \quad \forall j = 1 \dots n \quad (1.8') \\ \sum_{k=1}^{2^n+1} \mathbf{z}_{ik} = 1 \quad \forall i = 1 \dots m \quad (1.10) \\ \mathbf{z}_{ik} \in \{0; 1\} \end{array} \right.$$

- \mathbf{x}_i^k est le vecteur caractéristique du k -ième élément de l'ensemble Z_i , x_{ij}^k vaut 1 si ce groupe associé à \mathbf{z}_{ik} contient le client j , 0 sinon ;
- $c^{ik} = y_i^k f_i + \sum_{j=1}^n x_{ij}^k c_{ij}$ correspond donc au coût d'utilisation du groupe associé à \mathbf{z}_{ik} .

Il est possible de supprimer les variables correspondant à des équipements installés sans aucun client car elles ne seront jamais utilisées. Le vecteur nul faisant partie des points extrêmes de $\text{Conv}(Z_i)$, il est également possible de supprimer les variables correspondantes et de remplacer les contraintes de convexité (1.10) par des inégalités :

$$\sum_{k=1}^{2^n-1} \mathbf{z}_{ik} \leq 1 \quad \forall i = 1 \dots m$$

De plus, ici ces inégalités peuvent être retirées du modèle car elle seront systématiquement satisfaites à l'optimum. En effet, si une solution utilise une variable \mathbf{z}_{ik_1} correspondant à un groupe de clients J_1 et une variable une variable \mathbf{z}_{ik_2} correspondant à un groupe de clients J_2 , alors la solution utilisant uniquement la variable \mathbf{z}_{ik_3} correspondant au groupe de clients $J_1 \cup J_2$ présente un coût inférieur.

L'ensemble des variables $\{\mathbf{z}_k\}_{k=1 \dots m(2^n-1)}$ du modèle (LS_{set}) est égal à l'ensemble des variables associées aux éléments de l'ensemble $\bigcup_{i=1}^m (Z_i \setminus \{(\mathbf{0}, 0), (\mathbf{0}, 1)\})$. En in-

dexant les variables \mathbf{z}_{ik} à l'aide d'un seul indice au lieu de deux, on retombe donc sur le modèle (LS_{set}). De plus, les ensembles Z_i sont similaires à l'exception de l'indice i , ce qui fait que les m sous-problèmes peuvent être remplacé par un seul.

Cette modélisation sous forme de problème de couverture d'ensemble se fait de façon assez naturelle pour les problèmes de localisation-allocation, sans même avoir besoin d'explicitier l'utilisation de la décomposition de Dantzig-Wolfe.

1.3 Dualité lagrangienne et parallèle avec les méthodes de plans coupants

1.3.1 Dualité lagrangienne

Le lien entre le problème primal (ici le problème (\mathcal{P}) et sa version réduite (\mathcal{RP})) et son dual lagrangien s'établit par la relaxation lagrangienne. Dans la relaxation lagrangienne d'un problème, le respect des contraintes n'est plus obligatoire. En revanche, si l'une des contraintes est violée, le dépassement sera multiplié par un coefficient appelé multiplicateur de Lagrange, puis introduit dans la fonction objectif de façon à la pénaliser. Ces pénalités empêchent de trop s'écartier du domaine des solutions respectant les contraintes.

Reprenons l'exemple de notre problème (\mathcal{P}) . La contrainte (1.1) d'indice j de notre problème est violée lorsque :

$$\sum_{i \in S} a_{ij} z_i - b_j < 0$$

Nous allons multiplier l'écart de chaque contrainte (1.1) par un coefficient u_j positif et additionner les quantités ainsi obtenues. Il faut ensuite introduire cette quantité dans l'objectif pour pénaliser le non-respect des contraintes. Comme notre problème (\mathcal{P}) est une minimisation de fonction, il faut ajouter une quantité positive pour pénaliser l'objectif. Nous obtenons alors la fonction lagrangienne associée à notre problème :

$$l(\mathbf{u}, \mathbf{z}) = \sum_{i \in S} c_i z_i - \sum_{j=1}^n u_j \left(\sum_{i \in S} a_{ij} z_i - b_j \right) \quad (1.11)$$

Une fois définie la fonction lagrangienne, le dual lagrangien consiste alors à maximiser la fonction suivante :

$$(\mathcal{DL} - \mathcal{P}) \quad \max \mathcal{L}(\mathbf{u}) = \min_{\mathbf{z} \in \mathbb{R}_+^{|S|}} l(\mathbf{u}, \mathbf{z})$$

La fonction $l(\mathbf{u}, \mathbf{z})$ est bilinéaire donc, quel que soit $\mathbf{z}^0 \in \mathbb{R}_+^{|S|}$, $l(\mathbf{u}, \mathbf{z}^0)$ est une fonction linéaire de \mathbf{u} . $\mathcal{L}(\mathbf{u})$ est donc un minimum de fonctions linéaires. Par conséquent, la fonction $\mathcal{L}(\mathbf{u})$ est une fonction concave.

Chacune de ces fonctions linéaires peut être vue comme un hyperplan dans \mathbb{R}^{n+1} et la fonction $\mathcal{L}(\mathbf{u})$ représente alors l'enveloppe inférieure de tous les hyperplans.

Le dual de (\mathcal{P}) peut aussi s'écrire sous la forme d'un programme linéaire dual. Pour faire apparaître cela, on peut reformuler la fonction $l(\mathbf{u}, \mathbf{z})$ de façon à intervertir les

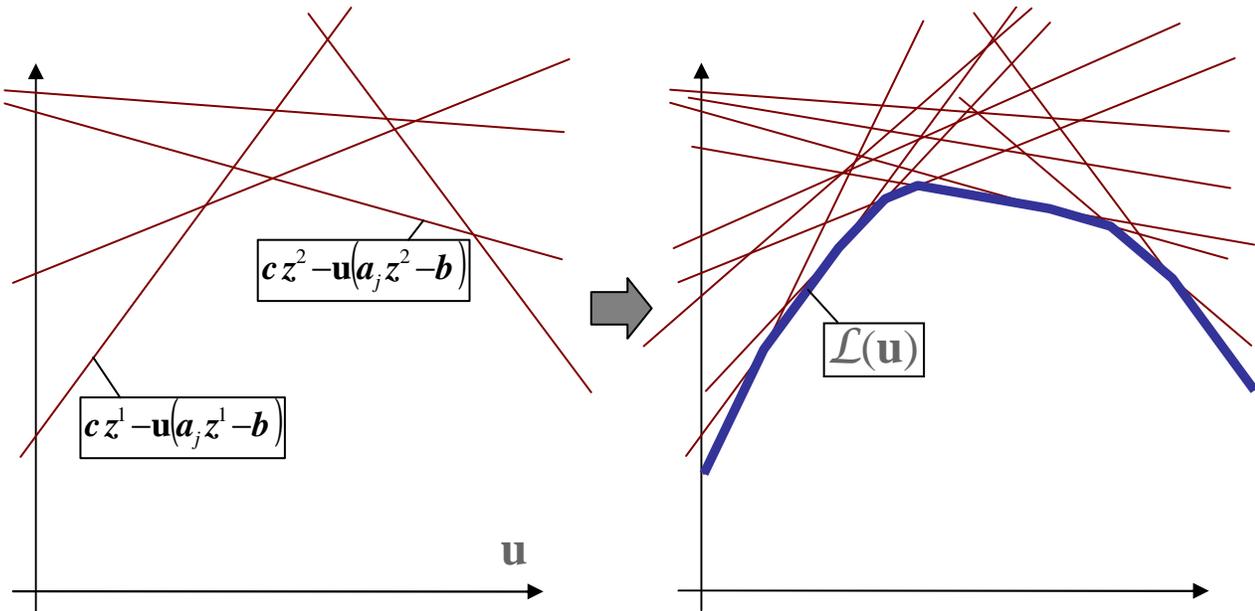


FIGURE 1.1 – Exemples d’hyperplans (ici des droites) correspondant à différentes valeurs du vecteur \mathbf{z} et représentation de la fonction de Lagrange

rôles des variables :

$$\begin{aligned}
 l(\mathbf{u}, \mathbf{z}) &= \sum_{i \in S} c_i \cdot \mathbf{z}_i - \sum_{j=1}^n \mathbf{u}_j \left(\sum_{i \in S} a_{ij} \mathbf{z}_i - b_j \right) \\
 &= \sum_{i \in S} c_i \cdot \mathbf{z}_i - \sum_{i \in S} \mathbf{z}_i \sum_{j=1}^n a_{ij} \mathbf{u}_j + \sum_{j=1}^n b_j \mathbf{u}_j \\
 &= \sum_{j=1}^n b_j \mathbf{u}_j - \sum_{i \in S} \mathbf{z}_i \left(\sum_{j=1}^n a_{ij} \mathbf{u}_j - c_i \right)
 \end{aligned} \tag{1.12}$$

Sous la forme (1.12) la fonction lagrangienne est associée au programme linéaire

suivant, qui est le programme linéaire dual de (\mathcal{P}) :

$$(\mathcal{D} - \mathcal{P}) \left\{ \begin{array}{l} \max_{\mathbf{u}} \quad \sum_{j=1}^n b_j \mathbf{u}_j \\ \text{s.c.} \\ \sum_{j=1}^n a_{ij} \cdot \mathbf{u}_j \leq c_i \quad \forall i \in S \\ \mathbf{u}_j \geq 0 \end{array} \right. \quad (1.13)$$

D'après le théorème de la dualité forte, nous savons que les valeurs optimales des problèmes $(\mathcal{DL} - \mathcal{P})$, $(\mathcal{D} - \mathcal{P})$, et (\mathcal{P}) sont identiques. Il est donc possible de résoudre le dual lagrangien de notre problème ou son programme linéaire dual plutôt que le problème lui-même. Malheureusement, pour le dual lagrangien, le nombre exponentiel de variables z_i rend impossible cette résolution, car il n'est pas possible de calculer facilement la quantité

$$\min_{\mathbf{z} \in \mathbb{R}_+^{|S|}} \left\{ \sum_{i \in S} c_i \cdot \mathbf{z}_i - \sum_{j=1}^n u_j \left(\sum_i a_{ij} \cdot \mathbf{z}_i - b_j \right) \right\}$$

pour toutes les valeurs de u . Quant au programme $(\mathcal{D} - \mathcal{P})$, c'est son nombre de contraintes qui rend sa résolution impossible.

1.3.2 Liens entre génération de colonnes et algorithmes de plans coupants

Plaçons-nous maintenant dans le cas où nous cherchons à résoudre la relaxation continue de $(\mathcal{DW} - \mathcal{Q})$ par génération de colonnes. Par commodité, supposons que \mathcal{X} est borné et donc que l'ensemble des rayons extrêmes de $\text{Conv}(\mathcal{X})$ est vide. La figure 1.2 nous montre les liens entre : la résolution par génération de colonnes de la décomposition de Dantzig-Wolfe du problème maître, la résolution de son dual lagrangien par l'algorithme de plans coupants de Kelley (voir [27]), et la résolution de son programme linéaire dual par génération de coupes. Penchons-nous d'abord sur la relation avec le dual lagrangien.

Première étape de la génération de colonnes : nous réduisons le nombre de variables pour obtenir le problème restreint $(\mathcal{DW} - \mathcal{RQ}_k)$. Ceci a pour effet de réduire également le dual lagrangien. Seul un sous-ensemble $I_k \subset P$ de points extrêmes est

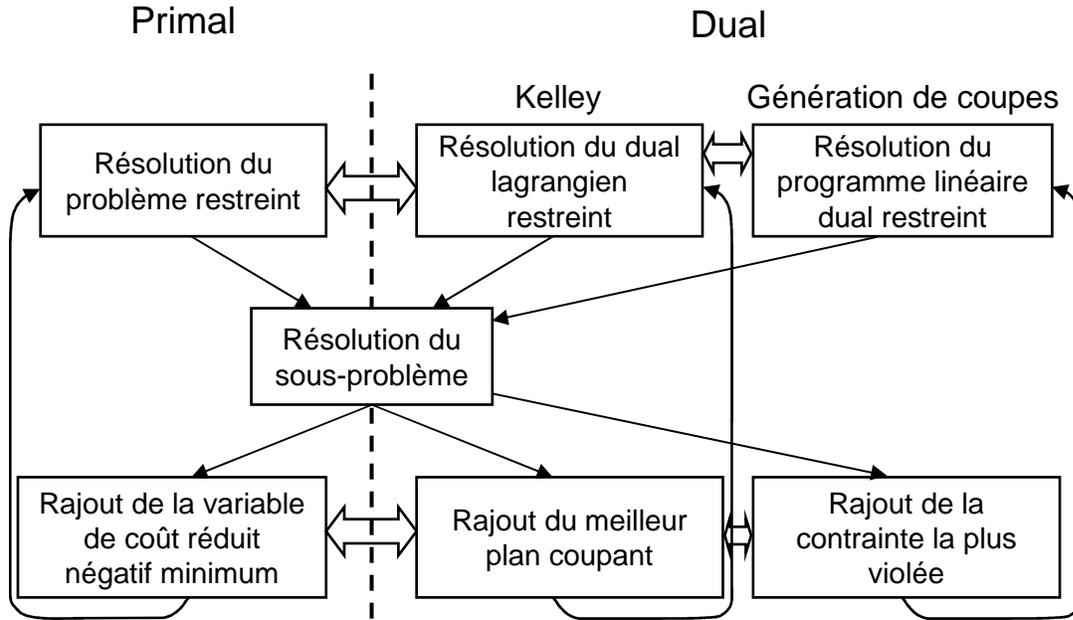


FIGURE 1.2 – Les étapes de la génération de colonnes et leurs contreparties du côté du dual, dans le cas d'une décomposition de Dantzig-Wolfe

considéré :

$$(\mathcal{DW} - \mathcal{RQ}_k) \left\{ \begin{array}{l} \min_{\lambda} \sum_{p \in I_k} \lambda_p c^p \\ \text{s.c.} \\ \sum_{p \in I_k} \lambda_p A x^p \geq \mathbf{b} \quad (1.3') \\ \sum_{p \in I_k} \lambda_p = 1 \quad (1.5) \\ \lambda_p \in \mathbb{R}^+ \forall p \in I_k \end{array} \right.$$

On effectue une relaxation lagrangienne de toutes les contraintes sauf la contrainte de convexité, et obtient donc à l'itération k le problème $(\mathcal{DL} - \mathcal{DW} - \mathcal{RQ}_k)$ suivant :

$$(\mathcal{DL} - \mathcal{DW} - \mathcal{RQ}_k) \max_{\mathbf{u} \in \mathbb{R}_+^n} \hat{\mathcal{L}}_{I_k}(\mathbf{u}) = \min_{\lambda} \left\{ \begin{array}{l} \sum_{p \in I_k} c^p \lambda_p - \mathbf{u} \cdot \left(\sum_{p \in I_k} \lambda_p A x^p - \mathbf{b} \right) \\ \text{s.c. : } \sum_{p \in I_k} \lambda_p = 1 \end{array} \right\}$$

La taille restreinte de l'ensemble I_k se traduit dans le dual lagrangien par l'absence de

certaines hyperplans. $\widehat{\mathcal{L}}_{I_k}(\mathbf{u})$ est donc une approximation supérieure de $\mathcal{L}(\mathbf{u})$. L'ajout de variables supplémentaires dans le problème primal, pendant la génération de colonnes, va rendre cette approximation de plus en plus juste, en rajoutant des hyperplans.

À chaque itération k de la génération de colonnes, $(\mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k)$ est résolu avec un nouvel ensemble de variables. Comme cela a été dit précédemment, la résolution de $(\mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k)$ nous apporte les valeurs optimales des variables duales u_j^k qui nous permettent de déterminer quelle variable rajouter dans $(\mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k)$. Mais ce vecteur \mathbf{u}^k est aussi une solution optimale du dual lagrangien restreint $(\mathcal{D}\mathcal{L} - \mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k)$. L'algorithme de plans coupants de Kelley consiste ici à rajouter un hyperplan tangent à $\mathcal{L}(\mathbf{u})$ en \mathbf{u}^k . Le sous-problème associé à $(\mathcal{D}\mathcal{L} - \mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k)$ s'exprime donc de la façon suivante :

$$\begin{aligned} & \min_{\lambda} \left\{ \begin{array}{l} \sum_{p \in P} c^p \lambda_p - \mathbf{u}^k \cdot \left(\sum_{p \in P} \lambda_p A \mathbf{x}^p - \mathbf{b} \right) \\ \text{s.c. : } \sum_{p \in P} \lambda_p = 1 \end{array} \right\} \\ &= \min_{\lambda} \left\{ \begin{array}{l} \sum_{p \in P} \left(c^p - \mathbf{u}^k \cdot A \mathbf{x}^p \right) \lambda_p + \mathbf{u}^k \cdot \mathbf{b} \\ \text{s.c. : } \sum_{p \in P} \lambda_p = 1 \end{array} \right\} \\ &= \min_{\lambda} \left\{ \begin{array}{l} \sum_{p \in P} \bar{c}^p \lambda_p \\ \text{s.c. : } \sum_{p \in P} \lambda_p = 1 \end{array} \right\} + \mathbf{u}^k \cdot \mathbf{b} \end{aligned}$$

où \bar{c}^p est le coût réduit associé à la variable λ_p . La contrainte de convexité fait que la solution de ce problème est le vecteur $\boldsymbol{\lambda}^k$ dont les coordonnées sont égales à 0 à l'exception de $\lambda_q = 1$, où $q = \arg \min_{p \in P} \{\bar{c}^p\}$. Trouver un hyperplan tangent à $\mathcal{L}(\mathbf{u})$ en \mathbf{u}^k revient donc à trouver la variable de coût réduit minimum.

Il est aussi important de noter que, maintenant que nous avons rajouté l'hyperplan correspondant à $\boldsymbol{\lambda}^k$, notre approximation de la fonction duale est exacte en \mathbf{u}^k :

$$\widehat{\mathcal{L}}_{I^{k+1}}(\mathbf{u}^k) = \bar{c}_q + \mathbf{u}^k \cdot \mathbf{b} = \mathcal{L}(\mathbf{u}^k)$$

$\widehat{\mathcal{L}}_{I^{k+1}}(\mathbf{u}^k)$ nous fournit donc une borne inférieure de la valeur optimale.

Intéressons-nous maintenant à la génération de coupes. Dans cette méthode, un programme linéaire dual restreint contenant seulement une partie des contraintes est

résolu à chaque itération. La résolution de ce programme linéaire à l'itération k donne un vecteur solution (\mathbf{u}^k, π^k) du dual (où π^k est la valeur duale optimale associée à la contrainte de convexité). La résolution du sous-problème pour ces valeurs des variables duales, permet de déterminer si il existe des contraintes du dual qui sont violées en (\mathbf{u}^k, π^k) et de rajouter la contrainte la plus violée. Le programme linéaire dual restreint de $(\mathcal{D}\mathcal{W} - \mathcal{Q})$ à l'itération k est le suivant :

$$(\mathcal{D} - \mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k) \begin{cases} \max_{\mathbf{u}, \pi} & \mathbf{u} \cdot \mathbf{b} + \pi \\ \text{s.c.} & \\ \mathbf{u} \cdot \mathbf{A}\mathbf{x}^p + \pi \leq c^p & \forall p \in I_k \\ \mathbf{u} \geq \mathbf{0}, \pi \in \mathbb{R} \end{cases} \quad (1.14)$$

Ici, le sous-problème consiste à trouver la contrainte la plus violée par le vecteur (\mathbf{u}^k, π^k) solution optimale de $(\mathcal{D} - \mathcal{D}\mathcal{W} - \mathcal{R}\mathcal{Q}_k)$, parmi l'ensemble de contraintes suivant : $\{\mathbf{u} \cdot \mathbf{A}\mathbf{x}^p + \pi \leq c^p \mid p \in P\}$. Une contrainte du dual est violée en (\mathbf{u}^k, π^k) lorsque le coût réduit de la variable correspondante dans le primal est négatif :

$$\begin{aligned} & \mathbf{u}^k \cdot \mathbf{A}\mathbf{x}^p + \pi^k > c^p \\ \Leftrightarrow & 0 > c^p - \mathbf{u}^k \cdot \mathbf{A}\mathbf{x}^p - \pi^k \\ \Leftrightarrow & \bar{c}^p - \pi^k < 0 \end{aligned}$$

Rechercher la contrainte la plus violée revient donc à résoudre : $\min_{p \in P} \bar{c}^p$, c'est à dire à chercher la colonne de coût réduit le plus négatif. Rajouter la contrainte la plus violée dans le dual correspond donc bien à rajouter la variable de coût réduit le plus négatif dans le primal.

1.4 Problèmes de stabilisation

Lors de la résolution d'un problème par la génération de colonnes, on constate souvent un ralentissement dans la convergence de la méthode. Ce ralentissement est souvent dû à des problèmes de stabilisation. Ils sont dus au fait que la suite des vecteurs \mathbf{u}^k des valeurs duales a du mal à converger vers les valeurs duales optimales du problème maître. Ceci nous oblige à devoir résoudre un grand nombre de sous-problèmes sans que les colonnes rajoutées n'apportent d'information réellement pertinente. Les figures 1.3 et 1.4 nous montrent deux exemples de problèmes de convergence des \mathbf{u}^k .

Dans le cas de la figure 1.3, les valeurs duales \mathbf{u}^k oscillent autour des valeurs à l'optimum. Ceci est dû au fait que notre approximation de la fonction $\mathcal{L}(\mathbf{u})$ n'est bonne

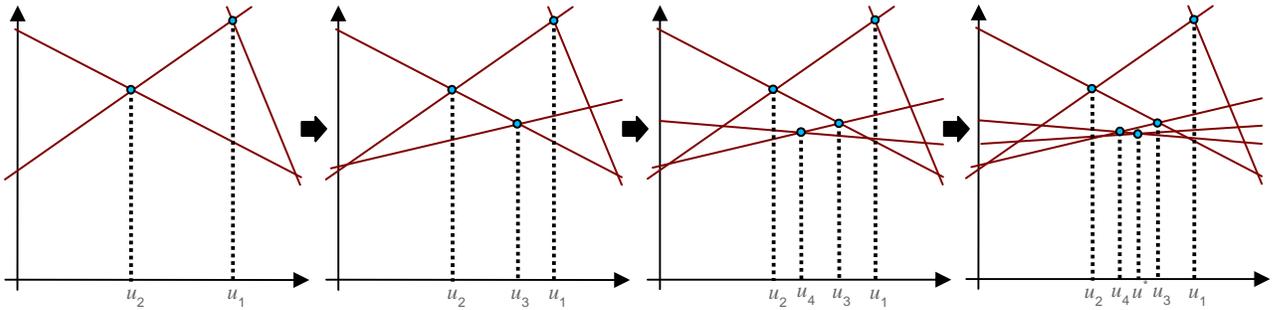


FIGURE 1.3 – Exemple de ralentissement de la convergence des valeurs duales

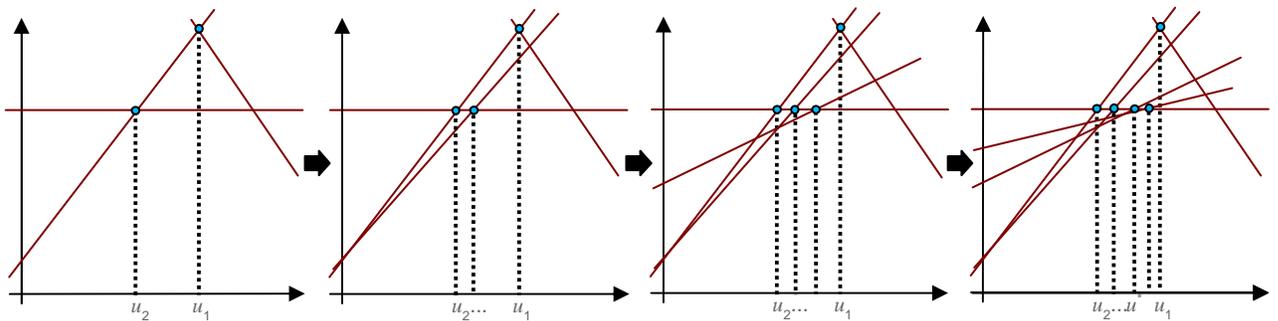


FIGURE 1.4 – Problème de convergence lié à la dégénérescence du primal

qu'à proximité des valeurs pour lesquelles nous avons résolu le sous-problème. Dès que la résolution du problème restreint nous amène à nous éloigner des \mathbf{u}^k précédemment utilisés, il y a un risque que le nouveau plan apporté par la résolution du sous-problème n'apporte que peu d'information utile.

Le problème de convergence peut aussi être dû à une dégénérescence du problème maître. Ceci se traduit notamment par la présence de nombreuses solutions optimales du dual. Dans notre représentation du dual lagrangien, cela correspondrait au rajout d'une droite parallèle à l'axe des abscisses, comme illustré dans la figure 1.4. Il se peut alors que pendant de nombreuses itérations, les droites rajoutées ne permettent pas d'améliorer la valeur de la solution actuelle du dual.

Plusieurs méthodes existent pour tenter de pallier à ces problèmes de stabilisation. Celles-ci sont l'objet de la prochaine section, notamment la méthode de boxstep et celle de stabilisation par point intérieur, utilisées dans les expérimentations des chapitres 4 et 5.

1.5 Méthodes de stabilisation

L'idée des méthodes de stabilisation est de donner de meilleures valeurs duales au sous-problème. Différentes méthodes apparaissent dans la littérature. Elles sont basées sur le fait que l'information apportée par les nouvelles colonnes n'est valable qu'à proximité du point de séparation où elles ont été trouvées. C'est à dire par exemple dans le cas du dual lagrangien, que l'hyperplan tangent à $\mathcal{L}(\mathbf{u})$ en \mathbf{u}^k ne fournit une bonne approximation de $\mathcal{L}(\mathbf{u})$ qu'à proximité de \mathbf{u}^k . Pour pallier ce problème, un premier groupe de méthodes, que j'appellerai méthodes de stabilisation proximales, cherchent à pénaliser les déplacements dans l'espace des solutions du dual. Pour cela, elles ajoutent dans l'objectif un terme qui sera fonction de l'éloignement par rapport à une valeur du vecteur dual \mathbf{u} considérée comme bonne. Plusieurs types de pénalité ont été étudiées dans la littérature. Cela va de l'imposition pure et simple de bornes sur les variables duales (méthode Boxstep), à l'ajout d'un terme quadratique dans l'objectif, en passant par des pénalités linéaires, linéaires par morceaux, ou encore un lissage entre \mathbf{u}^k et \mathbf{u}^{k+1} . Pour des détails sur ces différentes méthodes, voir [31]. Ces méthodes permettent de diminuer le nombre d'appels au sous-problème, mais parfois au détriment du temps de résolution du problème restreint. La méthode appelant le moins de fois le sous-problème peut donc ne pas être la plus rapide.

D'autres méthodes de stabilisation proposent de prendre un point de séparation qui soit à l'intérieur du polyèdre des solutions duales admissibles. Par exemple, l'algorithme « in-out » de [3] utilise comme point de séparation une combinaison convexe entre une solution admissible du dual non restreint, et une solution optimale du dual restreint. D'autres algorithmes de plans coupants comme l'algorithme ACCPM (également présenté dans [31]) ou l'algorithme d'Elzinga et Moore [17] changent la nature du problème maître pour trouver un centre (analytique ou géométrique) du polyèdre des solutions duales admissibles.

Dans [41], les auteurs proposent une méthode de stabilisation par point intérieur pour faire face aux problèmes de dégénérescence. Leur méthode consiste à prendre comme point de séparation une combinaison convexe de plusieurs points extrêmes du polyèdre des solutions duales optimales.

Ces différentes méthodes de stabilisation ne sont pas exclusives, il est possible d'en utiliser certaines conjointement. Dans les chapitres suivants nous utilisons notamment une méthode de boxstep avec la méthode de stabilisation par point intérieur de [41] que nous détaillons dans les sections suivantes.

1.5.1 Méthode Boxstep

La stabilisation par boxstep consiste à limiter les déplacements dans l'espace du dual en imposant des bornes aux valeurs duales. L'objectif est de limiter les oscillations

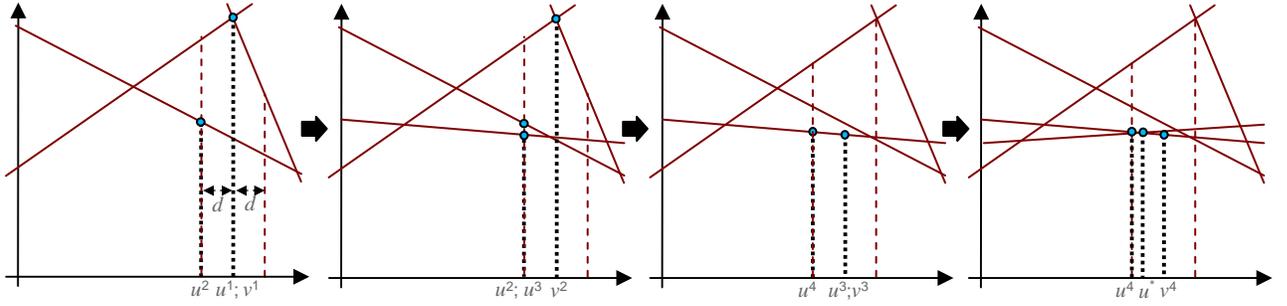


FIGURE 1.5 – Utilisation d'une méthode de boxstep avec une taille fixe d sur l'exemple de la figure 1.3

observées dans la figure 1.3 et d'assurer une convergence plus homogène. À chaque résolution du problème maître, les valeurs duales optimales sont contraintes à ne pas s'éloigner de plus d'une certaine distance d'un point du dual appelé centre de stabilité. Du point de vue du dual, tout se passe donc comme si le problème restreint était résolu à l'intérieur d'une « boîte ». Les contraintes de boîte rajoutées dans le dual à l'itération k ont la forme suivante :

$$\mathbf{u}_j^k \leq B_j^k + d^k \quad (1.15)$$

$$\mathbf{u}_j^k \geq B_j^k - d^k \quad (1.16)$$

Où B_j^k désigne la j -ème coordonnée du centre de stabilité à l'itération k et d^k représente la taille de la « boîte » à l'itération k . À l'itération k , le dual $(\mathcal{D} - \mathcal{RP}_k)$ du problème maître restreint a la forme suivante :

$$(\mathcal{D} - \mathcal{RP}_k \text{ boxstep}) \left\{ \begin{array}{l} \max_{\mathbf{u}} \quad \sum_{j=1}^n b_j \mathbf{u}_j \\ \text{s.c.} \\ \sum_{j=1}^n a_{ij} \mathbf{u}_j \leq c_i \quad \forall i \in I_k \quad (1.13) \\ \mathbf{u}_j^k \leq B_j^k + d^k \quad \forall j = 1 \dots n \quad (1.15) \\ \mathbf{u}_j^k \geq B_j^k - d^k \quad \forall j = 1 \dots n \quad (1.16) \\ \mathbf{u}_j \geq 0, \forall j = 1 \dots n \end{array} \right. \quad (1.13)$$

Ces contraintes dans le dual peuvent être intégrées au problème primal restreint sous la forme de colonnes artificielles. Dans le cadre de l'implémentation d'un algorithme

de génération de colonnes, cela permet d'éviter d'avoir à travailler à la fois sur le primal et le dual. Le problème restreint (\mathcal{RP}_k) devient alors le problème suivant :

$$(\mathcal{RP}_k \text{ boxstep}) \left\{ \begin{array}{l} \min_{\mathbf{z}, \mathbf{w}^+, \mathbf{w}^-} \sum_{i \in I_k} c_i \mathbf{z}_i + \sum_{j=1}^n \mathbf{w}_j^+ (B_j^k + d^k) - \mathbf{w}_j^- (B_j^k - d^k) \\ \text{s.c :} \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i + \mathbf{w}_j^+ - \mathbf{w}_j^- \geq b_j \quad \forall j = 1 \dots n \\ \mathbf{z}_i \geq 0 \quad \forall i \in I_k \\ \mathbf{w}_j^+, \mathbf{w}_j^- \geq 0 \quad \forall j = 1 \dots n \end{array} \right. \quad (1.17)$$

Ici, les variables \mathbf{w}^+ sont associées aux contraintes (1.15) et les variables \mathbf{w}^- aux contraintes (1.16). Une conséquence du rajout de ces variables artificielles \mathbf{w} est que la résolution du problème restreint $(\mathcal{RP}_k \text{ boxstep})$ n'apporte plus nécessairement une solution admissible du problème. En effet, une solution de $(\mathcal{RP}_k \text{ boxstep})$ respecte la contrainte (1.17), mais si les variables \mathbf{w}^+ ne sont pas nulles, le vecteur \mathbf{z} peut ne pas respecter les contraintes (1.1) du problème maître.

En revanche, dès que les valeurs duales optimales se trouvent à l'intérieur de la boîte (*i.e.* les inégalités (1.15) et (1.16) sont strictes), les conditions des écarts complémentaires imposent aux variables \mathbf{w} d'être nulles, et nous avons une solution admissible de (\mathcal{RP}_k) . Par conséquent, si les valeurs duales optimales de $(\mathcal{RP}_k \text{ boxstep})$ sont à l'intérieur de la boîte, et que le sous-problème ne trouve aucune colonne de coût réduit négatif, cela signifie que nous sommes à l'optimum du problème maître. De façon plus générale, lorsque le sous-problème ne trouve pas de variable de coût réduit négatif, nous sommes en présence d'une solution admissible du dual du problème maître, donc d'une borne inférieure de notre problème.

Pour que cette méthode soit efficace, il faut bien initialiser le centre de stabilité et la taille de la boîte, et choisir une bonne politique de mise à jour de ces deux éléments. L'initialisation du centre de stabilité peut se faire soit à l'origine, soit en utilisant une estimation des variables duales optimales. Attention toutefois, la qualité de la borne inférieure correspondant au centre de stabilité n'est pas toujours un bon critère pour son initialisation. Des vecteurs de valeurs duales fournissant une bonne borne peuvent se trouver très éloignés de la solution duale optimale. Plusieurs façons de mettre à jour le centre de stabilité sont proposées dans la littérature :

- Ne pas mettre à jour le centre de stabilité. Dans ce cas, seule la taille de la boîte varie.
- Mettre à jour le centre de stabilité à chaque itération k en utilisant les valeurs duales optimales du problème $(\mathcal{RP}_k \text{ boxstep})$.
- Mettre à jour le centre de stabilité lorsque le sous-problème ne retourne pas de variable de coût réduit négatif, toujours avec les valeurs duales optimales du problème $(\mathcal{RP}_k \text{ boxstep})$. Une variante consiste à ne mettre à jour le centre de stabilité

que lorsque les valeurs duales optimales permettent d'améliorer la borne inférieure du problème.

Pour ce qui concerne la taille de la boîte, il n'y a pas de règle bien définie. Dans les cas où l'on fait varier le centre de stabilité, on peut garder une taille fixe. Si la taille est trop grande, l'effet du boxstep sera négligeable. Mais une taille trop faible n'est pas efficace non plus, car cela peut accroître grandement le nombre d'itérations nécessaires pour atteindre l'optimum. De plus, une taille donnée ne conviendra pas à toutes les instances du problème. Par conséquent on trouve dans la littérature différentes politiques pour faire varier la taille de la boîte. On peut par exemple agrandir la taille lorsque le vecteur \mathbf{u}^k optimal est sur la limite de la boîte, ou bien l'agrandir uniquement lorsque le sous-problème ne trouve pas de colonne de coût réduit négatif et la réduire tant qu'il en trouve.

En limitant les déplacements dans l'espace dual, la méthode de boxstep permet d'éviter que la mauvaise qualité de l'approximation $\widehat{\mathcal{L}}_{I_k}$ en dehors du voisinage des vecteurs $\mathbf{u}^1 \dots \mathbf{u}^k$ n'ait trop d'impact sur la convergence. En revanche, cela n'a pas forcément d'effet sur les problèmes de dégénérescence. Pour lutter contre ces problèmes nous avons utilisé une autre méthode de stabilisation.

1.5.2 Méthode de stabilisation par point intérieur

Afin de limiter les problèmes de stabilisation dus à la dégénérescence, Rousseau *et al.* [41] proposent de choisir le point de séparation \mathbf{u}^k à l'intérieur de l'espace des solutions optimales du dual, et non comme un point extrême de ce dernier. Cela implique de résoudre plusieurs fois le problème restreint pour obtenir différents points extrêmes, puis de prendre la moyenne de ces points. Le temps de calcul du prochain terme \mathbf{u}^k est alors directement proportionnel au nombre de points extrêmes que l'on souhaite prendre en compte. Un compromis doit donc être trouvé entre l'augmentation de temps de calcul associé au problème restreint et le gain en nombre d'appels au sous-problème.

Les problèmes de dégénérescence se manifestent lorsque des colonnes de coût réduit négatif sont rajoutées dans le problème mais n'améliorent pas la fonction objectif. Cela se traduit dans l'espace dual par une infinité de solutions optimales. Par exemple, tout un segment peut-être solution optimale du dual, la valeur de la fonction objectif du dual étant la même sur tout ce segment. Dans ce cas, la résolution du problème restreint par l'algorithme du simplexe nous apporte le vecteur dual optimal \mathbf{u}^k associé à un point extrême de l'ensemble des solutions optimales du dual, *i.e.* à une extrémité du segment. Une colonne de coût réduit négatif sera trouvée par le sous-problème et la coupe correspondante dans le dual coupera l'extrémité du segment. On raccourcira ainsi le segment petit à petit jusqu'à ce qu'il disparaisse ou que l'on ne trouve plus de colonne de coût réduit négatif. Tant que ni l'un ni l'autre

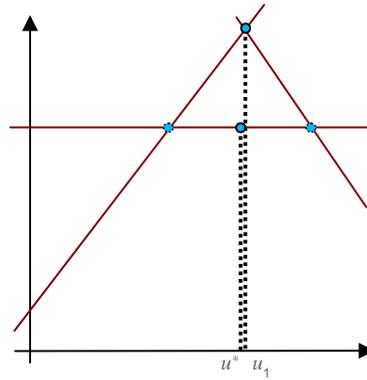


FIGURE 1.6 – Illustration du choix d'un point intérieur pour le vecteur dual \mathbf{u}^k

ne se produit, la valeur optimale ne variera pas, sans que l'on puisse conclure que l'optimum est atteint.

La méthode de stabilisation par point intérieur, introduite dans [41] propose d'accélérer la convergence en ne prenant plus pour \mathbf{u}^k une extrémité du segment, mais la moyenne de ses deux extrémités. De cette façon, nous serions assurés de réduire de moitié au moins la taille du segment. La figure 1.6 montre comment le prochain itéré \mathbf{u}^k est choisi. Elle est à comparer avec l'exemple de la figure 1.4.

En cas de dégénérescence, le polyèdre des solutions duales optimales a en général plus de dimensions qu'un segment, et plus de points extrêmes. Obtenir la totalité des points extrêmes est hors d'atteinte. Rousseau *et al.* proposent donc d'en générer un petit nombre en résolvant successivement des versions perturbées du programme linéaire dual. L'idée est d'identifier le polyèdre des solutions optimales duales associé à la solution optimale du problème primal restreint à l'aide des conditions des écarts complémentaires. On peut alors résoudre le problème dual sur ce polyèdre et affecter à chaque variable duale un coefficient dans la fonction objectif qui soit tiré aléatoirement. Pour chaque fonction objectif différente, on peut espérer obtenir ainsi un point extrême différent du polyèdre optimal.

Rappelons la forme du programme linéaire dual $(\mathcal{D} - \mathcal{RP}_k)$ de (\mathcal{RP}_k) :

$$(\mathcal{D} - \mathcal{RP}_k) \left\{ \begin{array}{l} \max_{\mathbf{u}} \quad \sum_{j=1}^n b_j \mathbf{u}_j \\ \text{s.c.} \\ \sum_{j=1}^n a_{ij} \mathbf{u}_j \leq c_i \quad \forall i \in I_k \quad (1.13) \\ \mathbf{u}_j \geq 0, \forall j = 1 \dots n \end{array} \right.$$

Notons \mathbf{z}^{k*} la solution optimale du problème (\mathcal{RP}_k) obtenue à l'itération k . Les conditions des écarts complémentaires pour le problème restreint (\mathcal{RP}_k) et son programme linéaire dual s'écrivent :

$$\left(\sum_{j=1}^n a_{ij} \mathbf{u}_j - c_i \right) \cdot \mathbf{z}_i^{k*} = 0 \quad (1.18)$$

$$\left(\sum_{i \in I_k} a_{ij} \mathbf{z}_i^{k*} - b_j \right) \cdot \mathbf{u}_j = 0 \quad (1.19)$$

En notant $J_k^* = \left\{ j / \sum_{i \in I_k} a_{ij} \mathbf{z}_i^{k*} - b_j > 0 \right\}$ et $I_k^* = \{ i / \mathbf{z}_i^{k*} > 0 \}$ on obtient les contraintes suivantes pour définir le polyèdre des solutions duales optimales associées à \mathbf{z}^{k*} :

$$\sum_{j=1}^n a_{ij} \mathbf{u}_j \leq c_i \quad \forall i \in I_k \setminus I_k^* \quad (1.20)$$

$$\sum_{j=1}^n a_{ij} \mathbf{u}_j = c_i \quad \forall i \in I_k^* \quad (1.21)$$

$$\mathbf{u}_j = 0 \quad \forall j \in J_k^* \quad (1.22)$$

$$\mathbf{u}_j \geq 0 \quad \forall j \in [1; n] \setminus J_k^* \quad (1.23)$$

Pour obtenir un autre point extrême de ce polyèdre que celui obtenu en résolvant

(\mathcal{RP}_k) , nous pouvons résoudre le problème suivant :

$$\left(\text{Modified } \mathcal{D} - \mathcal{RP}_k \right) \left\{ \begin{array}{l} \max_{\mathbf{u}} \quad \sum_{j=1}^n \omega_j \mathbf{u}_j \\ \text{s.c.} \\ \sum_{j=1}^n a_{ij} \mathbf{u}_j \leq c_i \quad \forall \quad i \in I_k \setminus I_k^* \quad (1.20) \\ \sum_{j=1}^n a_{ij} \mathbf{u}_j = c_i \quad \forall \quad i \in I_k^* \quad (1.21) \\ \mathbf{u}_j = 0 \quad \forall \quad j \in J_k^* \quad (1.22) \\ \mathbf{u}_j \geq 0 \quad \forall \quad j \in [1; n] \setminus J_k^* \quad (1.23) \end{array} \right.$$

Où ω_j est un réel choisi aléatoirement entre 0 et 1.

Dans ce programme linéaire, les contraintes nous assurent que l'on reste dans le polyèdre des solutions optimales du dual. Les coefficients des variables duales dans l'objectif ont été modifiés aléatoirement de façon à favoriser l'utilisation de telle ou telle variable, de sorte que les valeurs optimales diffèrent de celles obtenues en résolvant (\mathcal{RP}_k) . Pour obtenir des points extrêmes éloignés les uns des autres, on peut maximiser puis minimiser le problème ci-dessus.

La transposition des modifications de $(\text{Modified } \mathcal{D} - \mathcal{RP}_k)$ dans le primal donne le programme linéaire suivant :

$$\left(\text{Modified } \mathcal{RP}_k \right) \left\{ \begin{array}{l} \min_{\mathbf{z}} \quad \sum_{i \in I_k} (c_i \mathbf{z}_i) \\ \text{s.c.} \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i \geq \omega_j \quad \forall j \in [1; n] \setminus J^{k*} \quad (1.1'') \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i \geq -\infty \quad \forall j \in J^{k*} \quad (1.1''') \\ \mathbf{z}_i \geq 0 \quad \forall i \in I_k \setminus I^{k*} \\ \mathbf{z}_i \in \mathbb{R} \quad \forall i \in I^{k*} \end{array} \right.$$

Dans les contraintes (1.1''') le terme de droite vaut $-\infty$, ceci pour forcer les variables duales \mathbf{u}_j correspondantes à être nulles. Les contraintes de positivité sur les variables \mathbf{z}_i sont supprimées pour $i \in I^{k*}$ puisque les contraintes (1.21) dans le dual sont des égalités.

Comme présenté ci-dessus, les auteurs de [41] cherchent à obtenir différents points extrêmes du polyèdre des solutions duales optimales associées à une solution optimale précise \mathbf{z}^{k*} de (\mathcal{RP}_k) . Le polyèdre obtenu avec les conditions des écarts complémentaires dépend de cette solution \mathbf{z}^{k*} . Si (\mathcal{RP}_k) a plusieurs solutions optimales équivalentes, en ne considérant qu'une seule d'entre elles, on peut alors passer à côté de certaines solutions duales optimales. Il est donc peut-être préférable de chercher des points extrêmes du polyèdre de toutes les solutions optimales de $(\mathcal{D} - \mathcal{RP}_k)$. Pour obtenir ces points extrêmes, il est possible de rajouter simplement dans $(\mathcal{D} - \mathcal{RP}_k)$ une contrainte fixant la valeur de l'objectif et de modifier les coefficients de ce dernier de la même façon que dans $(\text{Modified } \mathcal{D} - \mathcal{RP}_k)$. Si on note F^* la valeur optimale de (\mathcal{RP}_k) , on pourrait substituer à $(\text{Modified } \mathcal{D} - \mathcal{RP}_k)$ le problème suivant :

$$(\text{Modified } \mathcal{D} - \mathcal{RP}_k \text{ bis}) \left\{ \begin{array}{l} \max_{\mathbf{u}} \quad \sum_{j=1}^n \omega_j \mathbf{u}_j \\ \text{s.c.} \\ \sum_{j=1}^n a_{ij} \mathbf{u}_j \leq c_i \quad \forall i \in I_k \quad (1.13) \\ \sum_{j=1}^n b_j \mathbf{u}_j = F^* \quad (1.24) \\ \mathbf{u}_j \geq 0 \end{array} \right.$$

Ces modifications sont assez simples à effectuer sur le programme linéaire dual, mais cela implique alors de devoir entretenir à la fois un modèle pour le primal et un modèle pour le dual au cours du déroulement de l'algorithme. Pour éviter cela, il est possible de transposer ces modifications dans le primal. En transposant la contrainte (1.24) dans le primal, on rajoute une variable dans (\mathcal{RP}_k) qui prend alors la forme suivante :

$$(\text{Modified } \mathcal{RP}_k \text{ bis}) \left\{ \begin{array}{l} \min_{\mathbf{z}, \phi} \quad \sum_{i \in I_k} (c_i \mathbf{z}_i) + F^* \phi \\ \text{s.c.} \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i + b_j \phi \geq \omega_j \quad \forall j = 1 \dots n \quad (1.1') \\ \mathbf{z}_i \geq 0 \quad \forall i \in I_k, \quad \phi \in \mathbb{R} \end{array} \right.$$

Il faudra prendre soin d'enlever la variable ϕ et de restaurer les coefficients de la fonction objectif une fois les points extrêmes alternatifs obtenus.

Lors de l'utilisation de solveurs comme CPLEX, les modifications des bornes des contraintes et des variables présentées dans $(\text{Modified } \mathcal{RP}_k)$ peuvent être aisément

réalisées à partir du modèle (\mathcal{RP}_k) . Selon le solveur utilisé, ces modifications peuvent être plus simples que de rajouter une variable au modèle pour l'enlever plus tard. C'est peut-être pour cette raison que Rousseau *et al.* ont préféré (*Modified \mathcal{RP}_k*) à (*Modified \mathcal{RP}_k bis*).

Le nombre l de vecteurs ω pour lesquels le problème (*Modified \mathcal{RP}_k*) sera résolu est le seul paramètre de la méthode. Ces résolutions apportent un ensemble de solutions optimales du dual $(\mathbf{u}^{k1}, \dots, \mathbf{u}^{kl})$ qui sont des points extrêmes du polyèdre optimal de $(\mathcal{D} - \mathcal{RP}_k)$. À partir de ces vecteurs, nous construisons le vecteur \mathbf{u}^{k*} de la façon suivante :

$$\mathbf{u}^{k*} = \sum_{i=1}^l \frac{\mathbf{u}^{ki}}{l} \quad (1.25)$$

Le vecteur \mathbf{u}^{k*} ainsi défini est utilisé comme point de séparation et passé au sous-problème.

1.6 Procédure de Branch & Price

La génération de colonnes permet de résoudre des problèmes continus. Les relaxations continues de problèmes entiers peuvent être résolues afin d'obtenir des bornes inférieures pour ces problèmes. Mais si l'on veut résoudre ces problèmes en nombre entiers, il faut intégrer la génération de colonnes dans un algorithme d'énumération implicite de type branch and bound. La méthode de résolution de problèmes en nombres entiers ainsi obtenue est appelée branch and price.

Le principe est d'effectuer une recherche arborescente dans l'espace des solutions. À la racine de l'arbre, la relaxation continue du problème est résolue par génération de colonnes. Si la solution est entière, inutile de chercher plus loin. Sinon, l'algorithme prend des décisions de branchements qui créeront plusieurs noeuds fils. Chacun des noeuds correspond à une branche différente de l'espace des solutions admissibles. Le but des décisions de branchement est de fixer petit à petit les variables fractionnaires pour essayer d'obtenir des solutions entières.

Chacun des noeuds fils sera résolu par génération de colonnes. Si la borne inférieure obtenue pour un noeud est supérieure à la meilleure solution entière connue, alors on peut couper ce noeud. Pour qu'un algorithme de branch and price soit efficace, il faut élaguer au maximum l'arbre des solutions. Cela passe par deux choses : obtenir de bonnes bornes ; construire un arbre équilibré. Ces deux points dépendent des décisions de branchement.

Dans les algorithmes classiques de branch and bound sans génération de colonnes, les décisions de branchement les plus communément utilisées consistent à brancher sur une seule variable fractionnaire. Malheureusement, pour les modèles adaptés à

la génération de colonnes, de telles règles de branchements peuvent produire des arbres très déséquilibrés. Par exemple, dans le cas d'une modélisation sous-forme de problème de couverture d'ensemble, la variable représentant la totalité de l'ensemble représente à elle seule une solution.

Dans le cas où l'on utilise la décomposition de Dantzig-Wolfe, on cherchera à brancher sur les variables \mathbf{x}_i du modèle initial. En notant $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ la solution optimale de la relaxation continue du noeud courant, on cherche $x_i^* = \sum_{p \in P} \lambda_p^* x_i^p + \sum_{r \in R} \mu_r^* x_i^r$ fractionnaire. Dans l'une des branches, on rajoutera la contrainte $\sum_{p \in P} \lambda_p x_i^p + \sum_{r \in R} \mu_r x_i^r \leq \lfloor x_i^* \rfloor$, et dans l'autre la contrainte $\sum_{p \in P} \lambda_p x_i^p + \sum_{r \in R} \mu_r x_i^r \geq \lceil x_i^* \rceil$. Ces contraintes peuvent être au choix ajoutées au problème maître ou intégrer de façon implicite au sous problème.

Pour les modélisations sous forme de couverture d'ensemble, une règle de branchement classique est celle de Ryan & Foster [42]. Elle consiste à effectuer le branchement sur les variables auxiliaires suivantes : $\sum_{k \in I} x_j^k x_i^k \mathbf{z}^k$ qui représentent le nombre d'ensembles couvrant à la fois les clients i et j dans la solution. L'idée est alors de créer une branche dans laquelle $\sum_{k \in I} x_j^k x_i^k \mathbf{z}^k = 0$ (qui signifie que les clients i et j doivent être couverts par des colonnes différentes), et une branche où $\sum_{k \in I} x_j^k x_i^k \mathbf{z}^k = 1$ (qui signifie que les clients i et j doivent être couverts par les mêmes colonnes).

D'autres aspects sont également importants lors de l'implémentation d'un algorithme de branch & price en dehors du choix des décisions de branchements : des aspects que l'on retrouve dans un branch & bound classique (l'implémentation d'heuristiques primales), ou des aspects spécifiques à la génération de colonnes (la gestion des colonnes, la possibilité de terminer la résolution de la relaxation continue avant d'atteindre l'optimum, ...). Pour une étude plus approfondie sur l'implémentation d'un Branch&Price, voir [45].

Chapitre 2

Génération de colonnes

« centrale »

Nous avons vu dans la section 1.3.2 que la génération de colonnes peut être considéré comme un algorithme de plans coupants du point de vue du dual. Or il existe de nombreux algorithmes de plans coupants différents. Nous allons maintenant nous intéresser à deux algorithmes de plans coupants dits « centraux ». Ils sont nommés ainsi car, au lieu de résoudre le sous-problème pour les valeurs optimales duales \mathbf{u}^{k*} du problème restreint, ils utilisent un vecteur de valeurs duales correspondant à un « centre » de l'ensemble des vecteurs duaux admissibles. Nous verrons ensuite comment utiliser l'un de ces algorithmes pour la génération de colonnes, et comment lui adapter les techniques de stabilisation classique détaillées dans les sections 1.5.1 et 1.5.2.

2.1 L'algorithme de plans coupants central d'Elzinga-Moore

Parmi les alternatives qui ont été proposées pour le choix du point du dual pour lequel sera résolu le sous-problème, les algorithmes centraux de plans coupants proposent de prendre un point central du polyèdre admissible, plutôt que d'en obtenir un point extrême en résolvant $(\mathcal{D} - \mathcal{RP}_k)$ avec l'algorithme du simplexe. Dans le cas de l'algorithme d'Elzinga et Moore [17], ce centre est géométrique.

Le principe de l'algorithme d'Elzinga et Moore est de choisir comme point de séparation le centre de la plus grande sphère contenue dans le polyèdre formé par les contraintes du dual auxquelles on ajoute une contrainte de borne sur la valeur de la solution (inférieure ou supérieure selon l'objectif du dual). L'utilisation de cette

méthode implique donc que le polyèdre soit borné. Cela est réalisable dans de nombreux problèmes en initialisant (\mathcal{RP}_0) avec des colonnes triviales (*i.e.* des colonnes faciles à construire, n'ayant pas forcément de chances de faire partie de la solution optimale).

Dans l'algorithme d'Elzinga et Moore, la résolution de $(\mathcal{D} - \mathcal{RP}_k)$ est donc remplacée par le problème (\mathcal{EM}_k) , qui cherche à maximiser le rayon σ de la sphère. Les contraintes (1.13) sont modifiées de la façon suivante :

$$\sum_{j=1}^n a_{ij} \mathbf{u}_j \leq c_i \quad \rightarrow \quad \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} \leq c_i \quad (2.1)$$

où \mathbf{v}_j est la j -ième coordonnées du centre de la sphère. Dans (2.1), la contrainte de droite s'assure que tout point de la sphère de coordonnées \mathbf{u} respecte la contrainte de gauche.

Pour la borne inférieure $(\mathcal{D} - \mathcal{RP}_k)$ est un problème de maximisation), notée LB , plusieurs possibilités existent pour la prendre en compte, avec des significations géométriques différentes.

$$\sum_{j=1}^n b_j \mathbf{v}_j - \sigma \geq LB \quad (2.2)$$

$$\sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} \geq LB \quad (2.3)$$

$$\sum_{j=1}^n b_j \mathbf{v}_j - \sigma(1 + \sqrt{\sum_{j=1}^n b_j^2 + 1}) \geq LB \quad (2.4)$$

Elzinga et Moore proposent la contrainte (2.2), qui indique que la valeur du dual pour le centre de la sphère est supérieure à la borne inférieure plus le rayon de la sphère. Betrò dans [4] propose de transformer la contrainte de borne inférieure $\sum_j b_j \mathbf{u}_j \geq LB$ au même titre que les contraintes (1.13). On obtient alors la contrainte

(2.3) qui assure que la valeur du dual en tout point de la sphère est supérieure à LB .

Pour obtenir la contrainte (2.4), il faut considérer une sphère de dimension $n + 1$. La dimension supplémentaire est celle de la valeur du dual. Le centre de la sphère a alors une coordonnée \mathbf{r} supplémentaire. Nous voulons que cette sphère soit contenue entre le plan de la borne inférieure et le plan de la fonction objectif de (\mathcal{RDP}_k) . Nous avons donc les deux contraintes suivantes :

$$\mathbf{r} + \sigma \sqrt{\sum_{j=1}^n b_j^2 + 1} \leq \sum_{j=1}^n b_j \mathbf{v}_j \quad (2.5)$$

$$\mathbf{r} - \sigma \geq LB \quad (2.6)$$

Il existe une sphère optimale pour laquelle l'inégalité (2.5) est saturée, c'est à dire une sphère qui touche le plan de la fonction objectif de $(\mathcal{D} - \mathcal{RP}_k)$. On peut donc remplacer \mathbf{r} dans (2.6) et on obtient la contrainte (2.4).

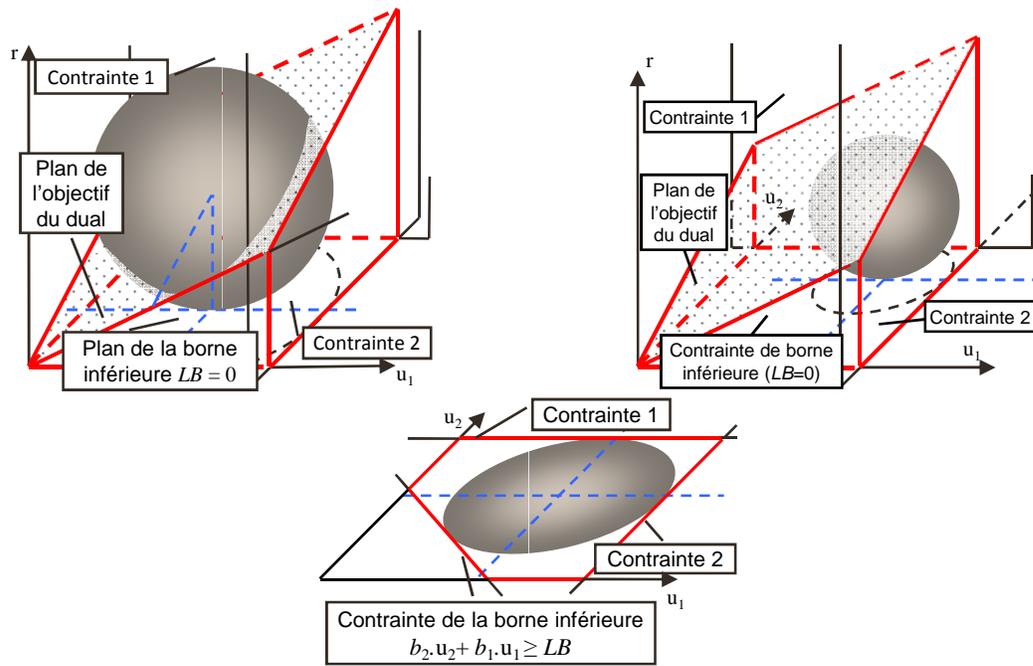


FIGURE 2.1 – Interprétation géométrique des trois inégalités de borne inférieure. À gauche l'inégalité 2.2, au centre 2.3, et à droite 2.4

La signification géométrique de la sphère et des trois inégalités de borne inférieure est illustrée par la figure 2.1. On peut voir que lorsqu'on a $\|\mathbf{b}\|_2 \geq 1$ (2.4) implique (2.3), qui implique (2.2). L'utilisation de (2.4) donne un polyèdre plus petit et le centre de la sphère sera plus proche de l'optimum de $(\mathcal{D} - \mathcal{RP}_k)$. Elle est donc aussi moins centrale donc potentiellement moins stable. Par la suite, nous avons opté pour l'inégalité 2.3, qui est un compromis entre les 3. Toutefois des expériences avec les autres formulations seraient nécessaires pour voir si elles apportent plus ou moins de stabilité. Le programme linéaire (\mathcal{EM}^k) résolu dans l'algorithme d'Elzinga-Moore est donc :

$$(\mathcal{EM}_k) \left\{ \begin{array}{l} \max_{\sigma, \mathbf{v}} \quad \sigma \\ \text{s.c.} \\ \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_{ij}^2} \geq LB \\ \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} \leq c_i \quad \forall i \in I_k \\ \mathbf{v}_j \geq 0 \quad \forall j = 1 \dots n ; \sigma \geq 0 \end{array} \right. \quad (2.3)$$

Pour éviter d'avoir à maintenir deux programmes linéaires (le primal (\mathcal{RP}_k) et le dual (\mathcal{EM}_k)) dans un algorithme de génération de colonnes, il peut être intéressant de dualiser (\mathcal{EM}_k) pour retourner dans l'espace primal. On obtient alors $(\mathcal{D} - \mathcal{EM}_k)$:

$$(\mathcal{D} - \mathcal{EM}_k) \left\{ \begin{array}{l} \min_{\mathbf{z}, \rho} \quad \sum_{i \in I_k} c_i \mathbf{z}_i - \rho LB \\ \text{s.c.} \\ \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_{ij}^2} \geq 1 \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho \geq 0 \quad \forall j = 1 \dots n \\ \mathbf{z}_i \geq 0 ; \rho \geq 0 \end{array} \right. \quad (2.7)$$

Le problème $(\mathcal{D} - \mathcal{EM}_k)$ présente peu de modifications par rapport à (\mathcal{RP}_k) : le rajout d'une variable, d'une contrainte, et la modification des termes de droite des inégalités (1.1). L'essentiel des coefficients des contraintes, à savoir les a_{ij} , ne changent pas, ce qui facilite l'utilisation d'une seule structure de données pour $(\mathcal{D} - \mathcal{EM}_k)$ et (\mathcal{RP}_k) dans l'implémentation.

Dans cette méthode, c'est donc le vecteur \mathbf{v}^{k*} , contenant les coordonnées optimales du centre de la sphère à l'itération k , qui est passé au sous-problème. Il se produit alors de 2 choses l'une :

- si le sous-problème trouve une variable de coût réduit négatif, elle est rajoutée à I_k qui devient I_{k+1} . Cette variable sera rajoutée à (\mathcal{EM}_k) sous la forme d'une nouvelle contrainte (2.1). On passe alors à l'itération $k + 1$.
- si le sous-problème ne retourne aucune variable de coût réduit négatif, alors \mathbf{v}^{k*} est une solution admissible du dual non restreint de (\mathcal{P}) . Donc, $\sum_{j=1}^n b_j v_j^{k*}$ est une borne inférieure de (\mathcal{P}) et la valeur de LB est mise à jour. On passe ensuite à l'itération $k + 1$.

À chaque itération de l'algorithme, la taille du polyèdre à l'intérieur duquel on recherche la sphère diminue. L'algorithme d'Elzinga & Moore s'arrête lorsque le rayon σ de la sphère est inférieur à un $\epsilon > 0$ donné. On peut alors repasser au primal et résoudre (\mathcal{RP}_k) pour obtenir une borne supérieure. Attention, ce critère d'arrêt ne signifie pas que la solution obtenue a une précision ϵ . Cette faiblesse du critère d'arrêt est corrigé dans l'algorithme de la section suivante, qui reprend le principe de celui d'Elzinga & Moore.

2.2 L'algorithme de plans coupants central accéléré de Betrò

Betrò reprend le principe de l'algorithme d'Elzinga & Moore en proposant principalement une accélération et un critère d'arrêt différent. À l'itération k de l'algorithme, étant données une borne inférieure LB^k et une borne supérieure UB^k pour notre problème, Betrò propose dans [4] de remplacer l'inégalité (2.3) dans (\mathcal{EM}^k) par

$$\sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} \geq \tau^k \quad (2.9)$$

où τ^k est une combinaison convexe de LB^k et de UB^k : $\tau^k = \alpha UB^k + (1-\alpha) LB^k$, $\alpha \in [0; 1[$. Le problème maître à résoudre dans l'algorithme de Betrò est alors le problème

(\mathcal{RBetro}_k) suivant :

$$(\mathcal{RBetro}_k) \left\{ \begin{array}{l} \max_{\sigma, \mathbf{v}} \quad \sigma \\ \text{s.c.} \\ \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} \geq \tau^k \quad (2.9) \\ \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} \leq c_i \quad \forall i \in I_k \quad (2.1) \\ \mathbf{v}_j \geq 0 \quad \forall j = 1 \dots n ; \sigma \geq 0 \end{array} \right.$$

τ^k agit comme une borne inférieure test. Ce n'est pas forcément une véritable borne inférieure de notre problème. L'algorithme se déroule donc ainsi :

À chaque itération k nous essayons de résoudre (\mathcal{RBetro}_k) .

- Si (\mathcal{RBetro}_k) est résolu, notons \mathbf{v}^{k*} le vecteur des coordonnées du centre de la sphère optimale. Nous résolvons alors le sous-problème pour \mathbf{v}^{k*} . Deux choses peuvent se produire, comme illustré dans la figure 2.2 :
 - le sous-problème retourne des colonnes de coût réduit négatif. Nous les rajoutons à I_k et passons à l'itération suivante.
 - le sous-problème ne retourne aucune colonne de coût réduit négatif. τ^k est une borne inférieure valide, et \mathbf{v}^{k*} est un point admissible du dual de (\mathcal{P}) . La valeur de LB est mise à jour avec $\sum_j b_j v_j^{k*}$, et τ^k est mis à jour également.
- Si (\mathcal{RBetro}_k) est infaisable, cela signifie que τ^k n'est pas une borne inférieure mais une borne supérieure de notre problème. UB est alors mise à jour ainsi que τ^k , comme illustré dans la figure 2.3. L'encadrement de la valeur optimale est donc multiplié par $\alpha < 1$:

$$\begin{aligned}
 UB^{k+1} - LB^{k+1} &= \tau^k - LB^k \\
 &= \alpha UB^k + (1 - \alpha) LB^k - LB^k \\
 &= \alpha (UB^k - LB^k)
 \end{aligned}$$

L'utilisation de τ^k , associée à un bon choix de α , permet de réduire l'encadrement de la valeur optimale bien plus rapidement qu'en utilisant l'algorithme d'Elzinga & Moore. De plus, le critère d'arrêt n'est plus basé sur la taille de la sphère mais sur la différence relative entre UB^k et LB^k . Ainsi, l'algorithme s'arrête pour une précision $\epsilon > 0$ donnée lorsque :

$$\frac{UB^k - LB^k}{UB^k} \leq \epsilon \quad (2.10)$$

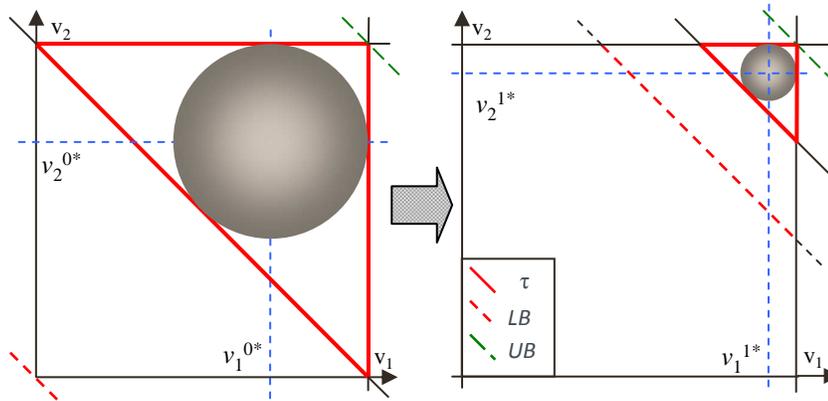


FIGURE 2.2 – Illustration du déroulement de l'algorithme de Betró lorsque (\mathcal{RBetro}_k) est faisable, pour $\alpha = 0,5$

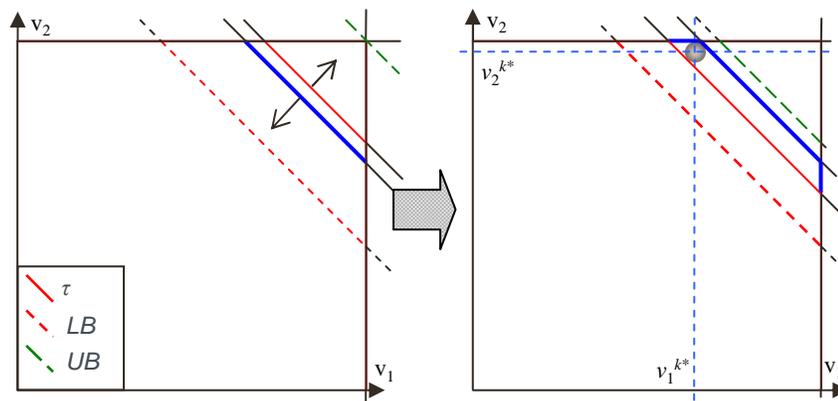


FIGURE 2.3 – Illustration du déroulement de l'algorithme de Betró lorsque (\mathcal{RBetro}_k) est infaisable, pour $\alpha = 0,5$

Tout comme pour l'algorithme d'Elzinga & Moore, il est possible de résoudre le dual $(\mathcal{D} - \mathcal{RBetro}_k)$ du problème (\mathcal{RBetro}_k) :

$$(\mathcal{D} - \mathcal{RBetro}_k) \left\{ \begin{array}{l} \min_{\mathbf{z}, \rho} \quad \sum_{i \in I_k} c_i \mathbf{z}_i - \tau^k \rho \\ \text{s.c.} \\ \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_j^2} \geq 1 \quad (2.11) \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho \geq 0 \quad \forall j = 1 \dots n \quad (2.12) \\ \mathbf{z}_i \geq 0 \quad \forall i \in I_k ; \rho \geq 0 \end{array} \right.$$

La résolution du problème $(\mathcal{D} - \mathcal{RBetro}_k)$ à la place de (\mathcal{RBetro}_k) implique un changement dans l'algorithme de Betrò. En effet, l'infaisabilité de (\mathcal{RBetro}_k) équivaut au fait que $(\mathcal{D} - \mathcal{RBetro}_k)$ soit non-borné. Si l'on utilise $(\mathcal{D} - \mathcal{RBetro}_k)$ comme problème maître, c'est lorsque celui-ci est non-borné que l'on saura que τ^k n'est pas une borne inférieure mais une borne supérieure.

Comme le montre Betrò dans [4], il est possible d'obtenir une solution admissible de (\mathcal{P}) à partir d'une solution de $(\mathcal{D} - \mathcal{RBetro}_k)$. Notons (\mathbf{z}^*, ρ^*) la solution optimale de $(\mathcal{D} - \mathcal{RBetro}_k)$. Si $\rho^* > 0$, alors :

$$\sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho \geq 0 \Rightarrow \sum_{i \in I_k} a_{ij} \frac{\mathbf{z}_i^*}{\rho^*} \geq b_j$$

Le vecteur $\frac{\mathbf{z}^*}{\rho^*}$ respecte donc les contraintes (1.1) et est une solution admissible de (\mathcal{P}) . Par contre, rien ne garantit que cette solution fournisse une amélioration de la borne supérieure.

2.3 Utilisation de l'algorithme central de plans coupants de Betrò dans une génération de colonnes

Les algorithmes centraux de plans coupants ont été développés dans le but d'accroître la stabilité des méthodes de plans coupants. La stabilité peut se montrer cruciale lors de la résolution d'un problème par génération de colonnes pour réduire l'effet de *tailing-off* (ralentissement de la convergence en approchant de l'optimum). D'où l'idée d'utiliser l'algorithme de Betrò pour une méthode de génération de colonnes et de le combiner avec les méthodes de stabilisation présentées dans les sections 1.5.1 et 1.5.2. Nous proposerons également une méthode de mise à jour du paramètre α et une modification pouvant permettre d'accélérer la terminaison de l'algorithme.

2.3.1 Choix du paramètre α et amélioration de la borne supérieure

Le paramètre α est celui qui détermine la progression de τ^k vers la borne supérieure UB . Un paramètre α peu élevé entraînera une diminution plus lente de la taille des polyèdres dans lesquels on cherche la sphère. La taille des sphères diminuera donc elle aussi plus lentement, et les points de séparation seront donc plus « centraux ». Cela s'est traduit par plus de stabilité dans nos expérimentations mais ralentit l'algorithme lorsque la stabilité n'est pas ou plus un problème.

Au contraire, un α élevé permet d'accélérer la convergence de τ^k vers UB . Les points de séparation se rapprocheront donc vite des solutions optimales du dual restreint classique ($\mathcal{D} - \mathcal{RP}_k$), et l'on gagnera peu en stabilité par rapport à la génération de colonne classique. Par contre, si UB est une très bonne borne (ou même la valeur optimale) et si la stabilité n'est pas ou plus un problème, un α élevé accélère la convergence vers une solution.

Betrò propose dans [4] d'utiliser un paramètre α constant pendant tout le déroulement de l'algorithme, et prend comme exemple un compromis avec $\alpha = 0,5$. Mais il est également possible de faire varier α . On retrouve ici des similarités avec le paramètre α de l'algorithme « in-out » de plans coupants présenté dans [3]. Dans l'algorithme « in-out », le point de séparation est une combinaison convexe d'un point admissible du dual non restreint et d'une solution optimale du dual restreint. Les mêmes arguments que ceux utilisés dans [3] permettent d'établir que la divergence de la série $\sum_k \alpha^k$, où α^k désigne la valeur du paramètre α à l'itération k , est une condition suffisante pour la convergence de l'algorithme.

Nous choisissons $\alpha_{min} \in]0; 1[\mid \forall k, \alpha^k \geq \alpha_{min}$ et $\alpha_{max} \in]0; 1[\mid \forall k, \alpha^k \leq \alpha_{max}$, et avons implémenté la politique de mise à jour des α^k suivante :

- Pour $k = 0$, α^0 est initialisé à α_{min} (par exemple, $\alpha_{min} = 0,15$) ;
- Lorsqu'à l'itération k , la borne inférieure LB est mise à jour, on effectue l'opération suivante : $\alpha^{k+1} = \alpha^k + (\alpha_{max} - \alpha^k)/10$;
- Lorsqu'à l'itération k , UB est mise à jour car τ est une borne supérieure, on effectue l'opération suivante : $\alpha^{k+1} = \alpha^k - (\alpha^k - \alpha_{min})/10$;
- Pour toutes les autres itérations, $\alpha^{k+1} = \alpha^k$.

Comme nous l'avons vu précédemment, lorsque UB est mise à jour, l'écart entre LB et UB est multiplié par α^k . Respectivement, lorsque LB est mise à jour, on a :

$$\begin{aligned} LB^{k+1} &> \tau^k = \alpha^k UB^k + (1 - \alpha^k) LB^k \\ \Rightarrow UB^{k+1} - LB^{k+1} &< (1 - \alpha^k)(UB^k - LB^k) \end{aligned}$$

Ainsi l'écart entre LB et UB est multiplié par un facteur inférieur à $(1 - \alpha^k)$. D'après

le critère d'arrêt (2.10) pour $\epsilon > 0$, et

$$\begin{aligned} \forall k \quad (1 - \alpha^k) &\leq (1 - \alpha_{min}) < 1 \\ \forall k \quad \alpha^k &\leq \alpha_{max} < 1 \end{aligned}$$

LB (respectivement UB) ne peut pas être mise à jour indéfiniment. Comme le nombre de colonnes est fini, l'algorithme converge.

Cette politique de mise à jour est basée sur le fait que, lorsque α vaut 0,5, la borne mise à jour est la plus éloignée de la valeur optimale. Nous cherchons à favoriser la borne la plus proche de la valeur optimale. C'est pour cette raison que nous pénalisons la borne qui est mise à jour.

La valeur du paramètre α_{min} est très faible pour permettre d'acquérir le plus de stabilité possible au début de l'algorithme. Pour la même raison, il pourrait être préjudiciable à la stabilité de l'algorithme d'initialiser LB à une valeur trop élevée. Lors de l'utilisation de l'algorithme dans une génération de colonnes, il est possible de mettre à jour la borne supérieure UB en calculant une solution admissible du problème (\mathcal{P}) que l'on cherche à résoudre. Nous avons vu dans la section précédente qu'il était possible de trouver une solution admissible de (\mathcal{P}) à partir d'une solution optimale de $(\mathcal{D} - \mathcal{RBetro}_k)$, mais que cela ne permettait pas forcément d'améliorer UB^k . Il est aussi possible de trouver une solution issue d'une heuristique, ou de la résolution du problème (\mathcal{RP}_k) . Ces solutions admissibles du primal peuvent permettre de mettre à jour plus souvent UB et ainsi d'accélérer l'algorithme. C'est pourquoi dans notre implémentation de l'algorithme, nous résolvons périodiquement le problème (\mathcal{RP}^k) . Si sa valeur optimale est inférieure à UB , alors nous mettons UB et τ^k à jour.

2.3.2 Adaptation de la méthode de Boxstep à l'algorithme de Betrò

Ici nous allons détailler l'adaptation d'une méthode de Boxstep à l'algorithme de Betrò. Rappelons que le principe du Boxstep est de limiter les déplacements dans l'espace dual entre chaque itération, en rajoutant des bornes sur les variables duales. Ces bornes forment une « boîte » centrée sur un point appelé centre de stabilité. Rappelons également que l'algorithme de Betrò utilise une borne inférieure supposée, et qu'il faut donc pouvoir déterminer si le problème maître est infaisable à cause de cette borne. Les contraintes de boîte viennent compliquer ce diagnostic. Pour voir cela, notons B_j^k les coordonnées du centre de stabilité et d^k la taille de la boîte à l'itération k . Les contraintes suivantes sont rajoutées à (\mathcal{RBetro}_k) :

$$\mathbf{v}_j \leq B_j^k + d^k \tag{2.13}$$

$$\mathbf{v}_j \geq B_j^k - d^k \tag{2.14}$$

On obtient alors le problème (\mathcal{RBetro}_k *boxstep*) :

$$(\mathcal{RBetro}_k \text{ boxstep}) \left\{ \begin{array}{l} \max_{\sigma, \mathbf{v}} \quad \sigma \\ \text{s.c.} \\ \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} \geq \tau^k \quad (2.9) \\ \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} \leq c_i \quad \forall i \in I_k \quad (2.1) \\ \mathbf{v}_j \leq B_j^k + d^k \quad \forall j = 1 \dots n \quad (2.13) \\ \mathbf{v}_j \geq B_j^k - d^k \quad \forall j = 1 \dots n \quad (2.14) \\ \mathbf{v}_j \geq 0 \quad \forall j = 1 \dots n ; \sigma \geq 0 \end{array} \right.$$

Les contraintes de boîte (2.13) et (2.14) peuvent rentrer en conflit avec la contrainte de borne inférieure (2.9) et rendre le problème (\mathcal{RBetro}_k *boxstep*) infaisable, alors même que le problème (\mathcal{RBetro}_k) sans contrainte de boîte reste faisable. Dans ce cas, l'infaisabilité du problème maître ne signifie pas que τ^k est une borne supérieure. Nous devons donc adapter l'algorithme de Betrò pour qu'en cas d'infaisabilité de notre problème maître avec *boxstep*, nous nous assurions que le problème maître sans *boxstep* est lui aussi infaisable. S'il l'est, τ^k est une borne supérieure et nous mettons à jour UB et τ^k . Si le problème sans *boxstep* est faisable, nous augmentons la taille de la boîte jusqu'à ce que le problème avec *boxstep* soit faisable.

Dans notre adaptation du *boxstep* à l'algorithme de Betrò, le centre de stabilité est mis à jour avec le centre de la sphère optimale lorsque le sous-problème ne retourne pas de variable de coût réduit négatif. L'augmentation de la taille de la boîte quant à elle se fait d'une façon un peu particulière. Lorsque le centre de stabilité est mis à jour, τ^k est augmenté, ce qui peut facilement entraîner l'infaisabilité de problème si la taille de la boîte est plus faible que l'augmentation de τ^k . Pour éviter cela, nous augmentons la taille de la boîte de la même quantité que τ^k à chaque fois que le centre de stabilité est mis à jour.

Voyons maintenant comment utiliser le *boxstep* dans le primal. La traduction des contraintes de boîte dans l'espace primal, donne le problème ($\mathcal{D} - \mathcal{RBetro}_k$ *boxstep*)

suivant :

$$(\mathcal{D} - \mathcal{RBetro}_k \text{ boxstep}) \left\{ \begin{array}{l} \min_{\mathbf{z}, \rho, \mathbf{w}^+, \mathbf{w}^-} \sum_{i \in I_k} c_i \mathbf{z}_i - \tau^k \rho + \sum_{j=1}^n \left(\mathbf{w}_j^+ \cdot (B_j^k + d^k) - \mathbf{w}_j^- \cdot (B_j^k - d^k) \right) \\ \text{s.c.} \\ \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_j^2} \geq 1 \quad (2.11) \\ \sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho + \mathbf{w}_j^+ - \mathbf{w}_j^- \geq 0 \quad \forall j = 1 \dots n \quad (2.15) \\ \mathbf{z}_i \geq 0 \quad \forall i \in I_k; \quad \rho \geq 0; \quad \mathbf{w}_j^+ \geq 0; \quad \mathbf{w}_j^- \geq 0 \end{array} \right.$$

où l'on retrouve les variables artificielles \mathbf{w}_j^+ et \mathbf{w}_j^- correspondant aux contraintes de boîte (voir section 1.5.1).

Tout comme cela est mentionné plus haut pour $(\mathcal{D} - \mathcal{RBetro}_k)$, le fait que τ^k ne soit pas une borne inférieure se traduira ici par le caractère non-borné du problème $(\mathcal{D} - \mathcal{RBetro}_k \text{ boxstep})$. Comme pour $(\mathcal{RBetro}_k \text{ boxstep})$, il faudra alors distinguer si cela est dû au boxstep ou à τ . Pour cela, il suffit d'examiner un rayon extrême responsable du caractère non-borné. Si une ou plusieurs variables artificielles \mathbf{w} apparaissent dans ce rayon extrême, alors les contraintes de boîte peuvent être responsables de l'infaisabilité du dual. Nous augmentons alors la taille de la boîte et tentons de résoudre à nouveau $(\mathcal{D} - \mathcal{RBetro}_k \text{ boxstep})$. Si le rayon extrême ne contient pas de variable \mathbf{w} , alors le boxstep n'est pas responsable du caractère non-borné de $(\mathcal{D} - \mathcal{RBetro}_k \text{ boxstep})$. Cela signifie que τ^k est une borne supérieure et nous pouvons mettre à jour UB et τ^k .

2.3.3 Adaptation de la méthode de stabilisation par point intérieur à l'algorithme de Betrò

Tout comme le problème $(\mathcal{D} - \mathcal{RP}^k)$, le problème (\mathcal{RBetro}_k) peut comporter une infinité de solutions optimales, à savoir plusieurs sphères de rayons identiques mais de centres différents. L'utilisation de l'algorithme de Betrò ne nous prémunit donc pas contre les problèmes de dégénérescence. Nous nous sommes donc intéressés à l'adaptation pour l'algorithme de Betrò de la méthode de stabilisation par point intérieur présentée dans la section 1.5.2. Le but de la stabilisation par point intérieur sera alors d'obtenir un point intérieur du polyèdre des centres des sphères optimales. Pour cela, il suffit de suivre les principes décrits dans [41].

Pour obtenir d'autres points extrêmes du polyèdre des centres optimaux, le plus simple est de rajouter une contrainte fixant la valeur du rayon à sa valeur optimale σ^* , et

de modifier les coefficients des variables \mathbf{v}_j dans l'objectif. Nous résoudrons donc des problèmes de la forme :

$$\left. \begin{array}{l}
 \max_{\sigma, \mathbf{v}} \quad \sum_{j=1}^n \omega_j \cdot \mathbf{v}_j \\
 \text{s.c.} \\
 \sigma = \sigma^* \\
 \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} \geq \tau^k \quad (2.9) \\
 \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} \leq c_i \quad \forall i \in I_k \quad (2.1) \\
 \mathbf{v}_j \geq 0 \quad \forall j = 1 \dots n ; \sigma \geq 0
 \end{array} \right\} \text{(Modified } \mathcal{RBetro}_k)$$

où les ω_j sont des coefficients tirés aléatoirement entre 0 et 1.

En résolvant plusieurs fois (*Modified* \mathcal{RBetro}_k) nous espérons obtenir plusieurs centres optimaux. Les valeurs duales passées au sous-problème seront alors les moyennes des coordonnées des centres trouvés.

Dans le cas où nous chercherions plutôt à résoudre le dual ($\mathcal{D} - \mathcal{RBetro}_k$) de (\mathcal{RBetro}_k), plusieurs possibilités s'offrent à nous :

– Première possibilité, résoudre le dual du problème (*Modified* \mathcal{RBetro}_k).

$$\left. \begin{array}{l}
 \min_{\mathbf{z}, \rho, \phi} \quad \sum_{i \in I_k} c_i \mathbf{z}_i - \tau^k \rho - \sigma^* \phi \\
 \text{s.c.} \\
 \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_j^2} - \phi \geq 0 \quad (2.16) \\
 \sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho \geq \omega_j \quad \forall j = 1 \dots n \quad (2.17) \\
 \mathbf{z}_i \geq 0 \quad \forall i \in I_k ; \rho \geq 0 ; \phi \in \mathbb{R}
 \end{array} \right\} \text{(Modified } \mathcal{D} - \mathcal{RBetro}_k)$$

– Deuxième possibilité, utiliser comme dans [41] les conditions des écarts complémentaires pour trouver le polyèdre des centres optimaux associés à une solution optimale (\mathbf{z}^*, ρ^*) de ($\mathcal{D} - \mathcal{RBetro}_k$). Les conditions des écarts complémentaires

pour le problème ($\mathcal{D} - \mathcal{RBetro}_k$) sont les suivantes :

$$\begin{aligned} \left(\sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} - \tau^k \right) \rho^* &= 0 \\ \left(\sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} - c_i \right) z_i^* &= 0 \\ \left(\sum_{i \in I_k} z_i^* \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho^* \sqrt{\sum_{j=1}^n b_j^2} - 1 \right) \sigma &= 0 \\ \left(\sum_{i \in I_k} a_{ij} z_i^* - b_j \rho^* \right) \mathbf{v}_j &= 0 \end{aligned}$$

Ce que nous cherchons, ce sont des vecteurs \mathbf{v} optimaux différents pour une même solution optimale du primal et pour une valeur optimale du rayon σ^* . Nous connaissons donc les z_i^* , ρ^* , et σ^* . Notons $I_k^* = \{i \in I_k \mid z_i^* > 0\}$ et $J_k^* = \{j \in [1; n] \mid \sum_{i \in I_k} a_{ij} z_i^* - b_j \rho^* > 0\}$. Les conditions des écarts complémentaires nous permettent de déduire le polyèdre des solutions duales optimales associé à la solution du primal z_i^*, ρ^* :

$$\begin{aligned} \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} &\geq \tau^k && \text{si } \rho^* = 0 \\ \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} &= \tau^k && \text{si } \rho^* > 0 \\ \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} &\leq c_i && \forall i \in I_k \setminus I_k^* \\ \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} &= c_i && \forall i \in I_k^* \\ \mathbf{v}_j &= 0 && \forall j \in J_k^* \\ \mathbf{v}_j &\geq 0 && \forall j \in [1; n] \setminus J_k^* \\ \sigma &= 0 && \text{si } \sum_{i \in I_k} z_i^* \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho^* \sqrt{\sum_{j=1}^n b_j^2} > 1 \\ \sigma &\geq 0 && \text{si } \sum_{i \in I_k} z_i^* \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho^* \sqrt{\sum_{j=1}^n b_j^2} = 1 \end{aligned}$$

La modification des coefficients des variables duales dans l'objectif, et la traduction de ces contraintes dans le primal, nous donne alors le problème (*Modified $\mathcal{D} - \mathcal{RBetro}_k$ bis*) :

$$\left(\text{Modified } \mathcal{D} - \mathcal{RBetro}_k \text{ bis} \right) \left\{ \begin{array}{l}
 \min_{\mathbf{z}, \rho} \sum_{i \in I_k} c_i \mathbf{z}_i - \tau^k \rho \\
 \text{s.c.} \\
 \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_j^2} \geq 0 \quad \text{si } \sum_{i \in I_k} z_i^* \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho^* \sqrt{\sum_{j=1}^n b_j^2} = 1 \quad (2.18) \\
 \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_j^2} \geq -\infty \quad \text{si } \sum_{i \in I_k} z_i^* \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho^* \sqrt{\sum_{j=1}^n b_j^2} > 1 \quad (2.18') \\
 \sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho \geq \omega_j \quad \forall j \text{ tel que } \sum_{i \in I_k} a_{ij} z_i^* - b_j \rho^* = 0 \quad (2.17) \\
 \sum_{i \in I_k} a_{ij} \mathbf{z}_i - b_j \rho \geq -\infty \quad \forall j \text{ tel que } \sum_{i \in I_k} a_{ij} z_i^* - b_j \rho^* > 0 \quad (2.17') \\
 \mathbf{z}_i \geq 0 \quad \forall i \mid z_i^* = 0 ; \quad \rho \geq 0 \text{ si } \rho^* = 0
 \end{array} \right.$$

Le choix entre $(\text{Modified } \mathcal{D} - \mathcal{RBetro}_k)$ et $(\text{Modified } \mathcal{D} - \mathcal{RBetro}_k \text{ bis})$ peut être tributaire des facilités d'implémentation fournies par le solveur choisi. Dans le premier, il faut pouvoir rajouter puis enlever facilement une variable, alors que dans $(\text{Modified } \mathcal{D} - \mathcal{RBetro}_k \text{ bis})$ il « suffit » de modifier les bornes des contraintes et des variables. L'utilisation de $(\text{Modified } \mathcal{D} - \mathcal{RBetro}_k)$ permet assez aisément de rajouter les variables artificielles du Boxstep comme nous allons le voir dans la section suivante.

2.4 Synthèse : algorithme central de génération de colonnes stabilisée

Les méthodes de Boxstep et de stabilisation par point intérieur ne sont pas exclusives. Il est tout à fait possible de rajouter des bornes sur les coordonnées du centre de la sphère dans le problème avec stabilisation par point intérieur. En ajoutant ces contraintes dans le problème $(\text{Modified } \mathcal{RBetro}_k)$, on obtient le programme

suisant :

$$\begin{array}{l}
 \left. \begin{array}{l}
 \max_{\sigma, \mathbf{v}} \quad \sum_{j=1}^n \omega_j \cdot \mathbf{v}_j \\
 \text{s.c.} \\
 \sigma = \sigma^* \\
 \sum_{j=1}^n b_j \mathbf{v}_j - \sigma \sqrt{\sum_{j=1}^n b_j^2} \geq \tau^k \\
 \sum_{j=1}^n a_{ij} \mathbf{v}_j + \sigma \sqrt{\sum_{j=1}^n a_{ij}^2} \leq c_i \quad \forall i \in I_k \\
 \mathbf{v}_j \leq B_j^k + d^k \quad \forall j = 1 \dots n \\
 \mathbf{v}_j \geq B_j^k - d^k \quad \forall j = 1 \dots n \\
 \mathbf{v}_j \geq 0 \quad \forall j = 1 \dots n ; \sigma \geq 0
 \end{array} \right\}
 \end{array}
 \quad \begin{array}{l}
 (2.9) \\
 (2.1) \\
 (2.13) \\
 (2.14)
 \end{array}$$

(Modified RBetro_k boxstep)

De même en ajoutant les variables artificielles correspondant aux contraintes de boîte dans le problème (*Modified D – RBetro_k*), on obtient le programme suivant :

$$\begin{array}{l}
 \left. \begin{array}{l}
 \min_{\mathbf{z}, \rho, \phi} \quad \sum_{i \in I_k} c_i \mathbf{z}_i - \tau^k \rho - \sigma^* \phi + \sum_{j=1}^n (\mathbf{w}_j^+ \cdot (B_j^k + d^k) - \mathbf{w}_j^- \cdot (B_j^k - d^k)) \\
 \text{s.c.} \\
 \sum_{i \in I_k} \mathbf{z}_i \sqrt{\sum_{j=1}^n a_{ij}^2} + \rho \sqrt{\sum_{j=1}^n b_j^2} - \phi \geq 0 \\
 \sum_{i \in I_k} a_{ij} \mathbf{z}_i - \rho + \mathbf{w}_j^+ - \mathbf{w}_j^- \geq \omega_j \quad \forall j = 1 \dots n \\
 \mathbf{z}_i \geq 0 \quad \forall i \in I_k ; \rho \geq 0 ; \phi \in \mathbb{R} ; \mathbf{w}_j^+ \geq 0 ; \mathbf{w}_j^- \geq 0
 \end{array} \right\}
 \end{array}
 \quad \begin{array}{l}
 (2.19) \\
 (2.20)
 \end{array}$$

(Modified D – RBetro_k boxstep)

Nous avons choisi dans notre implémentation d'un algorithme de génération de colonnes centrale de résoudre le problème (*Modified RBetro_k boxstep*) et de maintenir en parallèle le problème (*RP_k*) qui sera résolu périodiquement au cours de l'algorithme pour améliorer la borne supérieure, et également à la fin pour obtenir une solution admissible de (*P*).

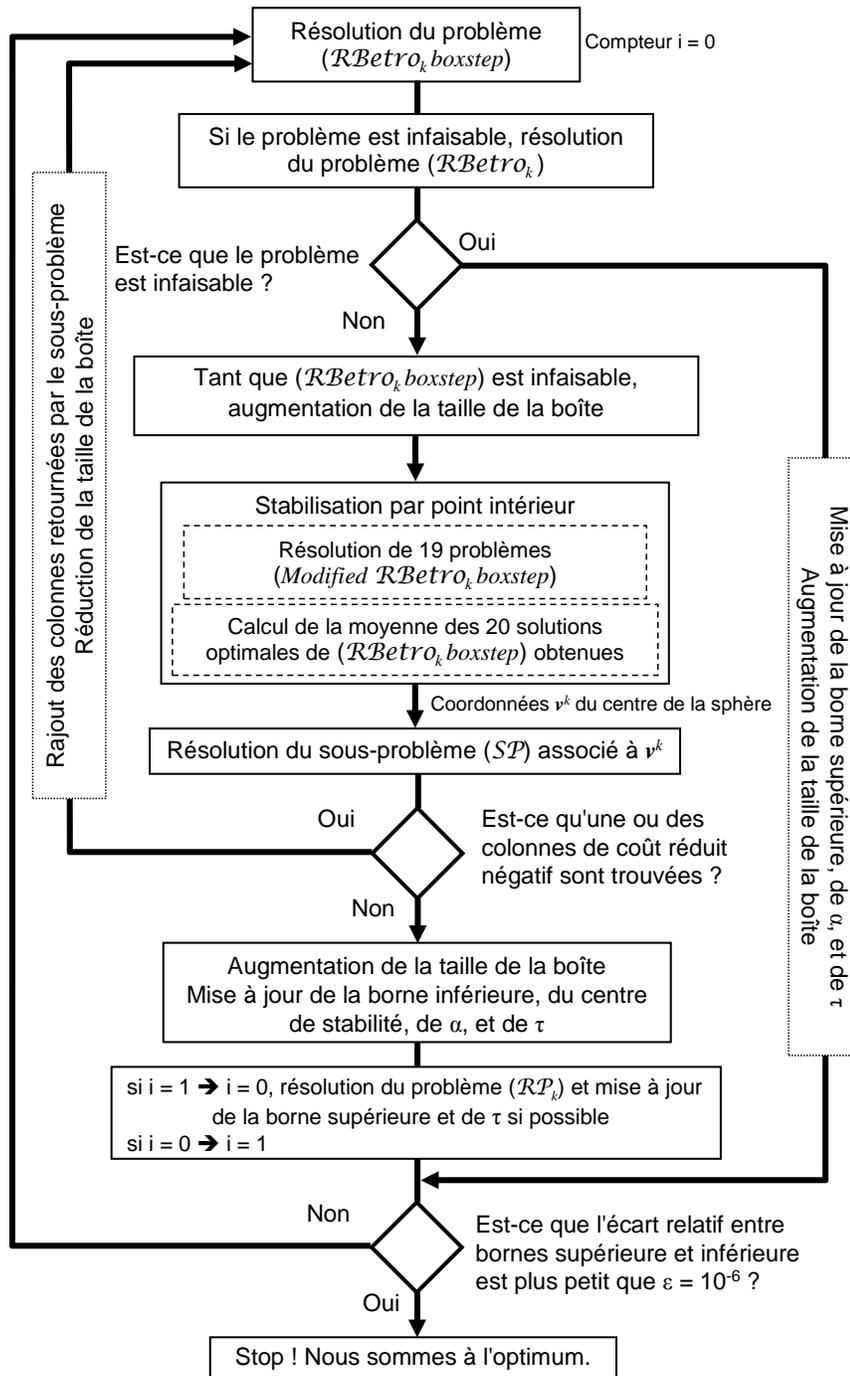


FIGURE 2.4 – Fonctionnement de la méthode de génération de colonnes centrale implémentée

	Instance	0	1	2	3	4	5	6	7	8	9	Moyenne
	Taille	171	170	176	176	180	198	179	197	160	153	176
Temps CPU Central (s)	Boxstep	1049	415	957	995	1031	2280	1370	1356	706	869	1189.7
	Point Intérieur	1478	565	1333	747	762	1154	704	1120	692	586	914.1
	Conjointement	207	136	173	181	170	195	145	183	162	121	167.3
Temps CPU Classique (s)	Boxstep	91	49	53	91	67	111	55	134	48	115	81.4
	Point Intérieur	375	302	385	437	463	714	571	810	272	210	453.9
	Conjointement	90	47	43	103	54	84	59	135	61	96	77.2

TABLE 2.1 – Temps de calcul des deux générations de colonnes avec différentes méthodes de stabilisation

Le déroulement de l'algorithme de génération de colonnes centrale implémenté à partir de l'algorithme de Betrò est détaillé dans la figure 2.4. Le paramètre concernant le nombre de problèmes résolus à chaque itération pour la méthode de stabilisation par point intérieur y est fixé à 20.

2.5 Comparaison des générations de colonnes classique et centrale

Dans cette section, nous comparons les bornes inférieures obtenue par génération de colonnes classique et par génération de colonnes centrale sur le problème de conception de réseau hiérarchique présenté plus loin dans le chapitre (5). Ce problème présente beaucoup de difficultés liées à la dégénérescence. Les mêmes méthodes de stabilisation (boxstep et stabilisation par point intérieur) sont utilisées pour les deux générations de colonnes, avec les mêmes paramètres dans la mesure du possible. 20 problèmes restreints sont résolus pour essayer de trouver des points extrêmes différents. Pour le boxstep, nous augmentons la boîte en multipliant sa taille par 1.5 (excepté quand τ^k est augmenté, comme expliqué dans la section 2.3.2). À chaque fois que le problème restreint est réalisable et que le sous-problème retourne des colonnes, la taille de la boîte est réduite en la multipliant par 0.9. Les initialisations du centre de stabilité, de la taille de la boîte, et des colonnes, sont les mêmes pour les deux méthodes.

En premier lieu, nous comparons l'efficacité des méthodes de stabilisation lorsqu'elles sont utilisées conjointement ou non. Le tableau présente les résultats des deux méthodes de générations de colonnes sur des petites instances. Les deux générations de colonnes sont utilisées séparément avec le boxstep et la stabilisation par point intérieur, et avec les deux techniques utilisées en même temps. Le tableau (2.1) présente les temps de calcul obtenus. Ces résultats nous montre qu'il est plus efficace d'utiliser

Instance	10	11	12	13	14	15	16	17	18	19	Moyenne
Taille	515	508	530	492	561	468	536	504	562	485	516.1
Temps CPU classique (s)	2951	1548	1196	2516	2971	1142	2118	2045	2059	1963	2050.9
Temps CPU centrale (s)	555	627	495	691	853	575	568	433	751	519	606.7
Instance	20	21	22	23	24	25	26	27	28	29	Moyenne
Taille	778	724	744	671	735	747	739	717	787	736	737.8
Temps CPU classique (s)	8981	5197	4569	4183	4702	11209	5536	3232	7906	4686	6020.1
Temps CPU centrale (s)	1162	702	1059	945	1138	1430	1084	1068	1336	1071	1099.5

TABLE 2.2 – Résultats numériques sur deux séries de dix instances

Instance	10	11	12	13	14	15	16	17	18	19	Moyenne
Temps CPU classique (s)	2403	1362	2718	2117	4069	1108	2306	2280	2650	1168	2218.1
Temps CPU centrale (s)	1022	623	673	455	3003	367	717	573	729	587	874.9

TABLE 2.3 – Temps CPU consommé avec une initialisation moins bonne

les deux méthodes de boxstep et de stabilisation par point intérieur conjointement. Ceci est particulièrement visible dans le cas de la génération de colonnes centrale. C'est moins évident pour la génération de colonnes classique où le gain en moyenne est assez faible. Nous pouvons également constater que sur ces petites instances, la génération de colonnes classiques est plus rapide que la génération de colonnes centrale. Les tests suivants sont tous effectués avec les deux méthodes de stabilisation utilisées conjointement.

Dans le tableau 2.2, sont présentés les résultats numériques obtenus sur deux ensembles d'instances de tailles moyennes différentes. Ces instances ont été obtenues à l'aide des outils de géométrie stochastiques présentés dans les sections 4.1.2 et 5.2.1. Les colonnes sont initialisées à partir de bonnes solutions heuristiques. Pour chaque instance sont notés sa taille, et le temps CPU utilisé par les deux méthodes de génération de colonnes.

Nous pouvons voir dans 2.2 que la génération de colonnes centrale a de meilleures performances que la génération de colonnes classique, avec des temps CPU moyens de 3 à 5 fois plus faible.

Dans le tableau 2.3, nous comparons le temps CPU utilisé sur le premier ensemble d'instances avec une initialisation moins bonne des colonnes. Les résultats montrent que cette absence d'information a peu d'impact sur les temps de calcul des deux méthodes.

Il est également intéressant de noter sur la figure 2.5 que la génération de colonnes centrale est surtout plus rapide que la génération de colonnes classique pour trouver

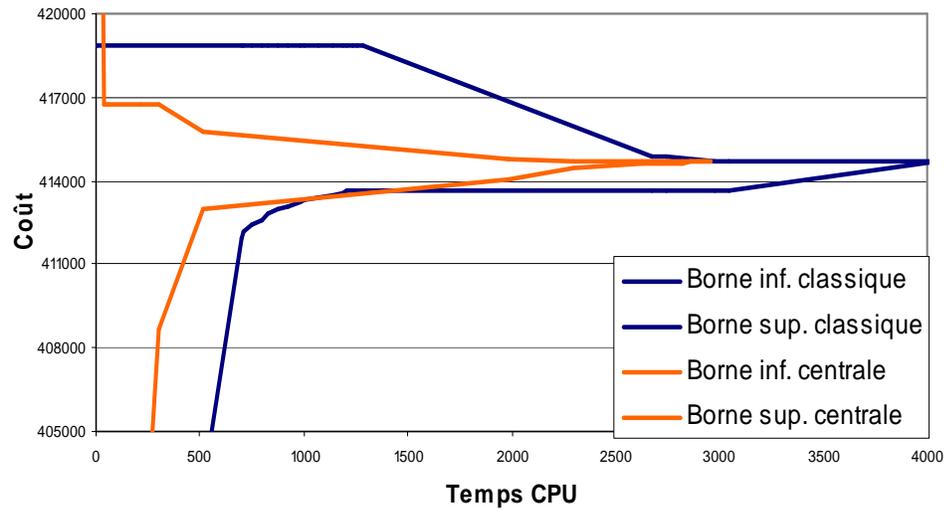


FIGURE 2.5 – Évolution des bornes sur l'instance 14 avec l'initialisation la moins bonne

des colonnes permettant d'améliorer la borne supérieure. Cela confirme le fait qu'elle est plus efficace contre les problèmes de dégénérescence.

2.6 Conclusion

Les expérimentations que nous avons menées sur un problème particulier de conception de réseau présentant de gros problèmes de dégénérescence montrent que notre implémentation d'une génération de colonnes centrale basée sur l'algorithme de plans coupants de Betrò est appropriée lorsque l'on a affaire à de gros problèmes de stabilisation. Elle s'est montrée bien plus performante qu'une génération de colonnes classique, même stabilisée, et nous pensons qu'à l'avenir elle montrera aussi son efficacité sur d'autres problèmes.

Chapitre 3

Problèmes de localisation dans les réseaux d'accès télécoms

Les problèmes de conception de réseau de télécommunications ont comme difficile tâche de répondre aux multiples questions qui se posent lors du déploiement d'un nouveau réseau :

- Quelle topologie générale pour mon réseau ? Combien de niveaux, quel types d'éléments ?
- Où placer les différents éléments ?
- Quelles capacités doivent avoir ces éléments ?
- Comment connecter les différents éléments du réseau (topologie, capacités des liens) ?

Les réponses à ces questions peuvent être liées à différents objectifs : minimisation du coût, fiabilité, réduction des délais, ... Élaborer un modèle répondant à l'ensemble de ces questions est très complexe. C'est pourquoi le plus souvent les problèmes de conception de réseau sont décomposés en problèmes s'occupant d'une partie spécifique du réseau. Ici nous nous intéressons à la conception de la partie accès du réseau, indépendamment du réseau cœur.

3.1 Problèmes de conception de réseau d'accès

Dans cette section nous abordons les multiples variantes du problème de conception de réseau d'accès (en anglais : *Local Access Network Design* ou *Local Access Telecommunication Network Design*). Dans ces problèmes, le but est d'acheminer un ensemble de demandes issues de terminaux, jusqu'à un équipement central relié au réseau cœur. De nombreux modèles cherchent à répondre aux questions du routage des demandes, du dimensionnement des liens, ou du placements de concen-

trateurs (des équipements qui compressent d'une façon ou d'une autre la demande pour diminuer le besoin en capacité en amont). Ces modèles respectent souvent des contraintes sur la topologie du réseau d'accès : en anneau, en étoile, en arbre, ou une combinaison de ces trois cas.

On trouve dans [8], [29], et [18], des études de différents modèles présentés dans la littérature. Dans [29, 8], les auteurs présentent aussi les différentes représentations du coût des liens : affine, concave linéaire par morceaux, linéaire par morceaux non concave, en escalier.

Dans [8], on trouve d'abord un modèle de localisation de concentrateurs très simple correspondant à un problème de localisation d'équipements avec capacité où chaque demande ne peut être affectée qu'à un seul concentrateur. Les coûts d'installation des concentrateurs diffèrent les uns des autres car ils prennent en compte le coût de la liaison jusqu'à l'équipement central. Les concentrateurs peuvent aussi bien représenter des changements de taille de câble apportant une économie d'échelle, que l'installation d'équipements proprement dit. Une variante de ce problème, appelée TELEPAK, consiste à considérer un nombre fini de types de concentrateurs différents pouvant être installés sur chaque site (ce qui reviendrait à proposer plusieurs tailles de câble pour assurer la liaison avec l'équipement central). Dans [44], les auteurs proposent de prendre en compte en plus une fonction du nombre de concentrateurs représentant l'économie d'échelle réalisée sur l'ensemble des concentrateurs achetés. Les auteurs utilisent des techniques de reformulation-linéarisation pour renforcer la relaxation continue du modèle et propose une heuristique à base de relaxation lagrangienne.

Les précédents modèles prennent en compte la capacités des concentrateurs et proposent des topologies de réseau étoile-étoile (*i.e.* les concentrateurs sont reliés à l'équipement central en formant une étoile, et les demandes assignées à chaque concentrateur forment elles aussi une étoile). Ils sont apparentés au problème classique de localisation d'équipements avec capacité (en anglais *Capacitated Facility Location Problem*) sur lequel nous reviendrons plus en détails dans la section 3.2.2.

D'autres modèles permettent d'envisager des topologies plus complexes. C'est le cas des modèles de conception de réseau d'accès, avec ou sans capacité. Dans ces derniers, le but est toujours de relier un ensemble de demandes à un équipement central, mais le graphe des liens utilisables n'a pas de topologie particulière, et les sommets de ce graphe n'ont pas forcément de demande. L'utilisation des liens présente un coût fixe et un coût variable dépendant de la demande.

Dans le cas avec capacité, on ne choisit pas uniquement quels arcs sont utilisés pour router les demandes, mais également la capacité à affecter à chacun. Les arcs peuvent par ailleurs disposer de capacités existantes. Notons qu'il n'est pas questions dans ces modèles de localiser des équipements. Dans [5], les auteurs étudient le polyèdre des solutions du problème et proposent plusieurs familles d'inégalités valides. Ces

inégalités sont utilisées pour renforcer la formulation du problème.

Dans le problème sans capacité, l'objectif est donc de choisir les liens utilisés et de router les demandes au sein de ces liens, de façon à minimiser le coût. Ce problème généralise le problème de l'arbre de Steiner et est donc NP-difficile. Dans [36], les auteurs proposent et comparent plusieurs méthodes de résolution pour ce problème. Une variante de ce problème modélisant un réseau d'accès utilisant deux technologies (la fibre optique et le cuivre) est étudiée dans [37]. Dans cette variante, les arcs « primaires » présentent un coût fixe plus élevé mais un coût variable plus faible, et forment une arborescence avec pour racine l'équipement central. Les demandes qui ne sont pas reliées à des arcs « primaires » seront desservies par des arcs « secondaires » qui forment des arborescences à partir de n'importe quel noeud desservi par un arc « primaire ». Le réseau d'accès obtenu a donc une topologie arbre-arbre. Il est à noter que les coûts d'équipements convertissant le signal entre les deux types d'arcs ne sont pas pris en compte.

D'autres variantes de modèles existent ne prenant en compte que le coût fixe des liens. C'est le cas par exemple du problème de conception de réseau hybride fibre-coaxial abordé dans [33]. Dans ce problème, les auteurs cherchent à raccorder les demandes à l'équipement central par l'intermédiaire d'équipements appelés ONU¹, faisant la transition entre un signal optique venant directement de l'équipement central, et un signal électrique sur câbles coaxiaux raccordés aux demandes. L'objectif consiste à localiser et dimensionner les ONU, et choisir la disposition des câbles coaxiaux pour raccorder chaque demande à un ONU. Les ONU sont raccordés de façon directe à l'équipement central. La topologie résultante est donc étoile-arbre. Les auteurs utilisent une heuristique ressemblant à une recherche tabou et basée sur une décomposition du problème en deux : un problème d'arbre couvrant minimum avec capacités puis un problème de localisation d'équipements avec capacités.

Pour modéliser des topologies de réseau complexes de type arbre-arbre ou présentant plusieurs niveaux, les problèmes sont souvent décomposés en plusieurs parties. Par exemple, dans [28] les auteurs proposent un problème de conception de réseau arbre-arbre basé sur le problème de conception de réseau d'accès sans capacité et prenant en compte en plus la localisation de concentrateurs (sans capacité). Pour le résoudre, les auteurs localisent les concentrateurs en résolvant un problème de localisation simple, puis des problèmes de conception de réseau d'accès sans capacité pour chacun des concentrateurs avec ses clients. Enfin un problème de conception de réseau d'accès sans capacité est résolu pour raccorder les concentrateurs à l'équipement central.

Dans [12], les auteurs présentent un modèle de conception de réseau hiérarchique qui traite à la fois de la topologie du réseau, du placement d'équipements, et des coûts fixes et variables des liens. Le modèle de programmation linéaire en nombres entiers proposé est valide pour plus de deux niveaux. C'est à dire qu'il peut prendre

1. *Optical Network Unit*

en compte plusieurs niveaux distincts de concentrateurs. Par contre le modèle ne prend pas en compte d'éventuelles capacités existantes. Les auteurs proposent un algorithme de branch&bound et utilise la relaxation lagrangienne pour calculer des bornes inférieures. Des résultats sont présentés pour seulement un niveau de concentrateurs et sur des réseaux ayant jusqu'à 100 noeuds et 400 arêtes.

Dans les sections suivantes sont présentés plus en détails les problèmes connus de localisation discrète et continue souvent utilisés dans l'optimisation des réseaux de télécommunications.

3.2 Problèmes de localisation discrète

Les problèmes de localisation discrète apparaissent souvent directement ou comme sous-problèmes dans l'optimisation des réseaux. Une étude les concernant ainsi que leur différentes variantes et applications dans les télécoms peut être trouvée dans [18]. Nous présentons dans cette section plus en détail deux de ces problèmes qui présentent des liens avec les applications des chapitres 5 et 6.

3.2.1 Problème de localisation simple

Le problème de localisation simple (en anglais *Simple Plant Location Problem* ou *Uncapacitated Facility Location Problem*) est un problème de localisation de concentrateurs sans capacité. Pour deux ensembles donnés de n clients et de m emplacements potentiels pour concentrateur, il s'agit de trouver le nombre et les emplacements des concentrateurs, ainsi que l'affectation des clients à ces derniers, de façon à minimiser la somme des coûts d'installation et d'affectation.

C'est ce problème qui a été utilisé dans la section 1.2.2. Nous rappelons ici le modèle mathématique et les notations :

$$(LS) \left\{ \begin{array}{l} \min_{\mathbf{x}, \mathbf{y}} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} \mathbf{x}_{ij} + \sum_{i=1}^m f_i \mathbf{y}_i \\ \text{s.c.} \\ \mathbf{x}_{ij} \leq \mathbf{y}_i \quad \forall i = 1 \dots m, \forall j = 1 \dots n \quad (1.6) \\ \sum_{i=1}^m \mathbf{x}_{ij} \geq 1 \quad \forall j = 1 \dots n \quad (1.7) \\ \mathbf{x}_{ij}, \mathbf{y}_i \in \{0, 1\} \end{array} \right.$$

où :

- m est le nombre d'emplacements potentiels pour les équipements ;
- n est le nombre de clients ;

- \mathbf{y}_i vaut 1 lorsqu'un équipement est placé sur le site i , 0 sinon ;
- \mathbf{x}_{ij} vaut 1 lorsque le client j est affecté à l'équipement installé sur le site i , 0 sinon ;
- c_{ij} est le coût d'affectation du client j au site i ;
- f_i est le coût d'installation d'un équipement sur le site i .

Le fait que les sites potentiels pour les concentrateurs soient fixés et l'absence de capacité (sur les concentrateurs comme sur les liens) permet d'intégrer les coûts fixes et variables des liaisons client-concentrateur dans les coefficients c_{ij} . De même, les coûts de liaison entre les concentrateurs et l'équipement central peuvent être inscrits dans les coefficients f_i . Le problème de localisation simple peut donc se concevoir comme un problème de conception de réseau étoile-étoile.

Le problème de localisation simple est NP-difficile (voir [11]), mais il a été beaucoup étudié et de nombreuses méthodes de résolution efficaces sont proposées dans la littérature : à l'aide de méthodes primal-dual [22], de relaxations lagrangiennes [2], ou encore de méta-heuristiques [19].

3.2.2 Problème de localisation d'équipements avec capacités

Lorsque l'on rajoute des capacités aux concentrateurs dans le problème précédent, on obtient le problème de localisation d'équipements avec capacité (en anglais *Capacitated Facility Location Problem*). Ce problème est également à la base de nombreux modèles d'optimisation de réseaux. Le principe est le même que celui du problème de localisation simple, sauf que les concentrateurs ne peuvent prendre en charge qu'une quantité limitée de demandes. Le problème est donc modélisé de la façon suivante :

$$(CFLP) \left\{ \begin{array}{l} \min_{\mathbf{x}, \mathbf{y}} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} \mathbf{x}_{ij} + \sum_{i=1}^m f_i \mathbf{y}_i \\ \text{s.c.} \\ \sum_{j=1}^n d_j \mathbf{x}_{ij} \leq q_i \mathbf{y}_i \quad \forall i = 1 \dots m \quad (3.1) \\ \sum_{i=1}^m \mathbf{x}_{ij} \geq 1 \quad \forall j = 1 \dots n \quad (3.2) \\ \mathbf{x}_{ij}, \mathbf{y}_i \in \{0; 1\} \end{array} \right.$$

où :

- m est le nombre d'emplacements potentiels pour les équipements ;
- n est le nombre de clients ;
- d_j est la demande du client j ;
- q_i est la capacité du concentrateur i ;
- \mathbf{y}_i vaut 1 lorsqu'un équipement est placé sur le site i , 0 sinon ;

- \mathbf{x}_{ij} vaut 1 lorsque le client j est affecté à l'équipement installé sur le site i , 0 sinon ;
- c_{ij} est le coût d'affectation du client j au site i ;
- f_i est le coût d'installation d'un équipement sur le site i ;
- Les contraintes (3.1) s'assurent que la demande totale affectée au concentrateur i n'excède pas sa capacité.

Ce problème a été abondamment étudié dans la littérature. Dans [18] sont présentés les derniers résultats notamment sur la structure polyédrique du problème.

3.3 Problèmes de localisation continue

Dans de nombreux problèmes de localisation liés à l'optimisation des réseaux, on dispose d'emplacements potentiels connus pour l'installation des nouveaux équipements. Toutefois, lors de l'installation d'un nouveau réseau sur un nouveau territoire, ou en l'absence d'infrastructure existante, ces emplacements prédéfinis n'existent plus. Dans ces cas là nous faisons face à des problèmes de localisation continue. Une étude de ces problèmes se trouve dans [14]. Nous présentons ici les problèmes classiques ayant un lien avec nos applications.

3.3.1 Le problème de Weber

Le problème de Weber (voir [14]) consiste à localiser un équipement dans le plan de façon à minimiser la somme des coûts de liaison de cet équipement à un ensemble de clients dont on connaît les coordonnées. Les coûts de liaisons en question sont en général proportionnels à la distance euclidienne, bien que d'autres distances puissent être utilisées.

Le problème de Weber est une minimisation de fonction convexe continue sans contrainte et l'algorithme le plus employé pour le résoudre est la procédure de Weiszfeld (voir [14]). Cette procédure itérative correspond en fait à une méthode de descente du gradient.

Le problème se présente classiquement sous cette forme :

$$\min_{\mathbf{x}, \mathbf{y}} \sum_{j=1}^n w_j \sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}$$

- n est le nombre de clients ;
- w_j est le poids du client j ;
- (\mathbf{x}, \mathbf{y}) sont les coordonnées de l'équipement ;
- (a_j, b_j) sont les coordonnées du client j .

La procédure de Weiszfeld est une procédure itérative correspondant à une méthode de descente du gradient à pas variable. La formule de calcul des itérations successives s'obtient à partir des conditions d'optimalité du premier ordre :

$$\begin{aligned} \sum_{j=1}^n \frac{w_j(\mathbf{x} - a_j)}{\sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}} = 0 & \quad \sum_{j=1}^n \frac{w_j(\mathbf{y} - b_j)}{\sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}} = 0 \\ \Leftrightarrow \frac{\sum_{j=1}^n \frac{w_j a_j}{\sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}}}{\sum_{j=1}^n \frac{w_j}{\sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}}} = \mathbf{x} & \quad \Leftrightarrow \frac{\sum_{j=1}^n \frac{w_j b_j}{\sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}}}{\sum_{j=1}^n \frac{w_j}{\sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2}}} = \mathbf{y} \end{aligned}$$

D'où :

$$\left(x^{(k+1)}, y^{(k+1)} \right) = \left(\frac{\sum_{j=1}^n \frac{w_j a_j}{\sqrt{(x^{(k)} - a_j)^2 + (y^{(k)} - b_j)^2}}}{\sum_{j=1}^n \frac{w_j}{\sqrt{(x^{(k)} - a_j)^2 + (y^{(k)} - b_j)^2}}}, \frac{\sum_{j=1}^n \frac{w_j b_j}{\sqrt{(x^{(k)} - a_j)^2 + (y^{(k)} - b_j)^2}}}{\sum_{j=1}^n \frac{w_j}{\sqrt{(x^{(k)} - a_j)^2 + (y^{(k)} - b_j)^2}}} \right) \quad (3.3)$$

Cette formule correspond à une méthode de descente du gradient pour un pas variable égal à $1 / \sum_{j=1}^n \frac{w_j}{\sqrt{(x^{(k)} - a_j)^2 + (y^{(k)} - b_j)^2}}$.

Comme la fonction objectif n'est pas différentiable aux coordonnées des clients, la procédure de Weiszfeld peut rencontrer un problème si une itération fait coïncider le point courant avec un client. Une façon d'éviter cet écueil consiste à utiliser une approximation des distances euclidiennes :

$$d_j(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - a_j)^2 + (\mathbf{y} - b_j)^2 + \epsilon} \quad \text{avec } \epsilon \text{ très petit}$$

De cette façon la fonction devient différentiable en tout point.

3.3.2 Le problème de Weber multi-source

Le problème de Weber multi-source a pour objectif de placer dans le plan continu un nombre donné, supérieur à un, d'équipements. Nous le présentons ici dans sa version classique sans capacité, qui possède donc les caractéristiques suivantes :

- Données :
 - Nombre d'équipements à placer ;
 - Nombre et emplacements des clients ;
 - Coûts par unité de distance des liaisons entre chaque équipement et chaque client.
- Sortie :
 - Raccordement (allocation) des clients ;
 - Localisation des équipements dans le plan continu.

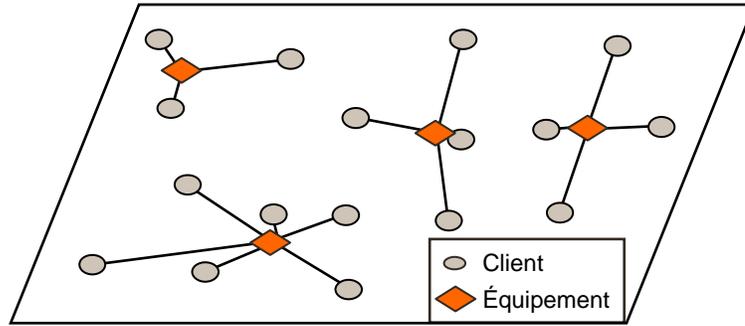


FIGURE 3.1 – Une solution d'un problème de Weber multi-source

Tout comme pour le problème de Weber, la distance la plus souvent utilisée est la distance euclidienne mais d'autres distances peuvent aussi être employées. La différence avec le problème de Weber vient donc du fait d'avoir plusieurs équipements à placer, ce qui fait passer le problème dans le domaine de l'optimisation combinatoire. Dans le cas de la distance euclidienne ce problème est particulièrement difficile car la fonction objectif n'est ni convexe ni concave et peut présenter un grand nombre d'optimums locaux. Dans [30], les auteurs ont montré qu'il était NP-Difficile en le formulant comme l'énumération des partitions en cellules de Voronoï (voir définition page 74) de l'ensemble des clients.

En notant \mathbf{x}_{ij} la variable représentant la fraction de la demande du client j satisfaite par l'équipement i , et $(\mathbf{r}_i, \mathbf{s}_i)$ les coordonnées de l'équipement i , la formulation mathématique du problème de Weber multi-source dans le cas de la distance euclidienne est la suivante :

$$(\mathcal{PWM}) \left\{ \begin{array}{l} \min_{\mathbf{X}, \mathbf{r}, \mathbf{s}} \sum_{i=1}^m \sum_{j=1}^n \mathbf{x}_{ij} w_j \sqrt{(\mathbf{r}_i - a_j)^2 + (\mathbf{s}_i - b_j)^2} \\ \text{s.c.} \\ \sum_{i=1}^m \mathbf{x}_{ij} = 1 \quad \forall j = 1 \dots n \quad (3.4) \\ \mathbf{x}_{ij} \in \{0, 1\} ; \mathbf{r}, \mathbf{s} \in \mathbb{R}^m \end{array} \right.$$

Avec :

- m nombre d'équipement ;
- n nombre de clients ;
- \mathbf{x}_{ij} fraction de la demande du client j satisfaite par l'équipement intermédiaire i ;
- w_j est le poids du client j ;
- $(\mathbf{r}_i, \mathbf{s}_i)$ sont les coordonnées de l'équipement i ;
- \mathbf{X} la matrice des \mathbf{x}_{ij} ;

- \mathbf{r} le vecteur des \mathbf{r}_i ;
- \mathbf{s} le vecteur des \mathbf{s}_i .

Ce problème étant NP-difficile, de nombreuses heuristiques ont été développées pour le résoudre. Dans [6], les auteurs effectuent une comparaison de plusieurs de ces heuristiques. Des méthodes de génération de colonnes pour résoudre ce problème de façon exacte ont également été développées (voir [23] et [39]). La variante du problème de Weber multi-source où le nombre d'équipements à placer est inconnu a également été étudiée dans la littérature. La résolution de cette variante est l'objet du chapitre 4.

Le problème de Weber Multi-source classique est un problème de localisation de concentrateurs. Il est nécessaire de le modifier quelque peu pour en faire un problème de conception de réseau d'accès, notamment en rajoutant les liaisons entre concentrateurs et équipement central. C'est le sujet de l'application du chapitre 5.

3.3.3 Le problème de Weber avec réseau de transport alternatif

Le problème de Weber avec réseau de transport alternatif reprend le principe du problème de Weber en permettant en plus d'utiliser un réseau existant sur la totalité ou une partie des liaisons entre l'équipement et les clients. Ce réseau présente un avantage au niveau du coût (sinon il ne serait pas utilisé) et il est représenté par un graphe. Cette variante de problème de Weber a notamment été étudiée dans [34] et [9].

Dans [34], les auteurs proposent un modèle de programmation linéaire en nombre entier. Pour cela, ils s'appuient sur une modification du graphe de transport alternatif. Introduisons les notations suivantes :

- Soit $G = (V, E)$ le graphe de transport, avec $V = \{v_1, \dots, v_o\}$ l'ensemble des sommets et E celui des arêtes.
- Soit $A = \{a_1, \dots, a_n\}$ l'ensemble des clients.
- Soient $d(a_n, v_p)$ la distance euclidienne entre le client n et le sommet p , $k(v_p, v_q)$ la longueur de l'arête (p, q) , et $d_G(v_p, v_q)$ la longueur d'un plus court chemin entre v_p et v_q dans G .

Un graphe G' est donc construit à partir du graphe G avec les transformations suivantes :

- Les sommets du graphe G' sont les éléments de l'ensemble $I = V \cup A$.
- Tout élément a_m de $A \setminus V$ est raccordé à tout élément v_p de V par une arête de longueur $d(a_m, v_p)$ (*i.e.*, tous les clients qui ne sont pas des sommets du graphe G sont raccordés à tous les sommets du graphe G).
- Si on a $k(v_p, v_q) \geq d(v_p, v_q)$, la longueur de l'arête (v_p, v_q) est remplacée par $d(v_p, v_q)$.
- Si on a $d(v_p, v_q) < d_G(v_p, v_q)$, une nouvelle arête (v_p, v_q) de longueur $d(v_p, v_q)$ est rajoutée .

Dans le graphe G' , on tient du fait qu'il soit parfois plus intéressant de se rendre d'un sommet du graphe G à un autre de manière directe dans le plan plutôt que par un chemin du graphe G . Par exemple, il peut être plus intéressant de se rendre à pied d'une station de métro à une autre si celle-ci sont sur deux lignes différentes qui n'ont pas de correspondance à proximité.

Il est possible de montrer que la longueur d'un plus court chemin entre deux sommets de G' correspond à la longueur d'un plus court chemin entre ces deux sommets dans le plan muni du graphe G . Grâce à cela, le chemin entre l'équipement à localiser et un client va pouvoir être représenté uniquement par un *point intermédiaire* représentant le premier sommet de G' traversé par le chemin. Ce point peut être le client lui-même. Dans le cas où l'on cherche à localiser un seul équipement, on note i_1, \dots, i_k l'ensemble des sommets de G' et \mathbf{y}_{rm} la variable de décision qui vaut 1 si le client a_m est raccordé grâce au *point intermédiaire* i_r , 0 sinon. On obtient alors la formulation suivante du problème :

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & \sum_{m=1}^n w_m \left(\sum_{r=1}^k \mathbf{y}_{rm} (d(\mathbf{x}, i_r) + d_{G'}(i_r, a_m)) \right) \\ \text{s.c.} \quad & \\ & \sum_{r=1}^k \mathbf{y}_{rm} = 1 \quad \forall m = 1, \dots, n \\ & \mathbf{x} \in \mathbb{R}^2 \\ & \mathbf{y}_{rm} \in \{0, 1\} \quad \forall r = 1, \dots, k, m = 1, \dots, n \end{aligned} \tag{3.5}$$

où :

- w_m est le poids du client m ;
- \mathbf{x} est le point du plan où est localisé l'équipement ;
- $d(\mathbf{x}, i_r)$ est la distance entre l'emplacement de l'équipement \mathbf{x} et le sommet i_r de G' ;
- $d_{G'}(i_r, a_m)$ est la longueur du plus court chemin dans G' entre le sommet i_r et le client a_m (elle peut être nulle si $i_r = a_m$).

Le problème (3.5) pourrait être utilisé dans l'optimisation des réseaux d'accès pour localiser un concentrateur dans le plan en présence d'une infrastructure existante (une voirie ou des conduites ou poteaux). Toutefois, ce problème est plus complexe que celui purement discret qui consisterait à localiser l'équipement sur le graphe G' . On peut alors se demander si la précision gagnée sur la localisation du concentrateur en vaut la peine.

Chapitre 4

Résolution du problème de Weber multi-source avec nombre d'équipements inconnu

La variante du problème de Weber Multi-source où le nombre d'équipements est inconnu et le coût d'installation d'un équipement constant a déjà été étudié dans la littérature, par exemple dans [10] et [7]. Cette variante du problème de Weber multi-source peut se formuler sous la forme :

$$(WM\acute{e}quipements) \left\{ \begin{array}{l} \min_{\mathbf{X}, \mathbf{r}, \mathbf{s}, \mathbf{m}} \sum_{i=1}^{\mathbf{m}} \sum_{j=1}^n \mathbf{x}_{ij} w_j \sqrt{(\mathbf{r}_i - a_j)^2 + (\mathbf{s}_i - b_j)^2} + \mathbf{m} C \\ \text{s.c.} \\ \sum_{i=1}^{\mathbf{m}} \mathbf{x}_{ij} = 1 \quad \forall j = 1 \dots n \quad (4.1) \\ \mathbf{x}_{ij} \in \{0, 1\} ; \mathbf{r}, \mathbf{s} \in \mathbb{R}^{\mathbf{m}} ; \mathbf{m} \in \mathbb{N}^* \end{array} \right.$$

où C est le coût d'installation d'un équipement.

Jusqu'à récemment les approches pour résoudre ce problème supposaient que le coût optimal hors coûts d'installation était une fonction convexe du nombre d'équipements. Les méthodes de résolutions du problème de Weber multi-source classique étaient donc utilisées pour des valeurs successives du nombre d'équipements jusqu'à obtenir un minimum local. Non seulement cette approche n'est pas très efficace puisqu'elle remplace un problème NP-Difficile par $O(n)$ problèmes de même difficulté, mais en plus, il est possible de trouver des contre-exemples à l'hypothèse de convexité faite au départ, comme montré dans [7]. Pour résoudre le problème de façon approchée, l'idée est alors d'obtenir une bonne estimation du nombre d'équipements dans

la solution optimale, puis d'employer les mêmes techniques que pour le problème de Weber multi-source. Nous verrons qu'il est également possible d'utiliser la génération de colonnes sur le problème avec nombre d'équipements inconnus, comme sur le problème de Weber multi-source classique.

La variante du problème de Weber Multi-source où le nombre d'équipements est inconnu et le coût d'installation d'un équipement constant, est très proche du problème de télécommunication étudié dans le chapitre 5. Ici nous avons deux niveaux hiérarchiques : les équipements et les clients. Dans l'application du chapitre 5 il y a trois niveaux hiérarchiques : les clients, les équipements intermédiaires, et un équipement supérieur auquel tous les équipements intermédiaires sont rattachés. Les coûts changent également. La liaison entre un équipement intermédiaire et l'équipement supérieur a un coût fixe dépendant uniquement de la distance entre les deux équipements, et également un coût variable qui dépend aussi du nombre de clients rattachés à l'équipement intermédiaire.

Malgré ces différences, nous avons pu adapté à notre application des méthodes utilisées pour le Weber multi-source avec nombre d'équipements inconnu : l'heuristique multi-phase de [7], une méthode de recuit simulé, et l'approche de génération de colonnes. De même, nous avons adapté au Weber multi-source les approches de résolution de notre application utilisant la géométrie stochastique. Nous présentons donc dans ce chapitre l'ensemble de ces différentes méthodes de résolution et leur résultats sur ce problème de localisation classique. Dans le chapitre suivant nous retrouverons ces mêmes méthodes dans le cadre de notre application industrielle.

4.1 Estimation du nombre optimal d'équipements

4.1.1 Résolution du problème de localisation simple associé

Cooper, dans [10], a remarqué sur des instances précises de problèmes de Weber multi-source avec distance euclidienne que l'emplacement optimal d'un équipement est souvent très proche de l'un des clients, quand il n'est pas confondu avec l'un d'eux. Une façon d'obtenir un raccordement proche de l'optimal, pourrait donc consister à résoudre le problème en considérant les emplacements des clients comme étant les seuls emplacements potentiels pour les équipements. C'est l'idée de l'heuristique présentée dans [10]. Toutefois, dans cet article, Cooper énumère l'ensemble des allocations pour toutes les valeurs possibles du nombre m d'équipements pour résoudre ce problème discrétisé. Il est possible de faire mieux. En fait, Cooper résout pour chaque valeur de m l'équivalent discret du problème de Weber multi-source, appelé p -médian, en énumérant toutes les solutions admissibles. L'idée de résoudre un problème p -médian pour obtenir un bon raccordement est reprise dans [20] où les auteurs présentent une heuristique pour le problème de Weber multi-source. Cette

transformation du problème continu en problème discret s'appuie sur le comportement asymptotique du nombre optimal d'équipements quand le nombre de clients tend vers l'infini. Plus la densité de clients augmente, plus les équipements seront placés près d'un des clients, et plus les raccordements obtenus pour les problèmes discret et continu seront proches.

En gardant cette idée de discrétiser le problème [7] propose de résoudre un problème de localisation simple pour le cas du problème de Weber multi-source avec nombre d'équipements inconnu. Les emplacements potentiels pour les équipements sont ceux des clients. Ce problème classique de localisation discrète se présente sous la forme suivante :

$$(LS) \left\{ \begin{array}{l} \min_{\mathbf{X}, \mathbf{y}} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} \mathbf{x}_{ij} + \sum_{i=1}^n f_i \mathbf{y}_i \\ \text{s.c.} \\ \mathbf{x}_{ij} \leq \mathbf{y}_i \quad \forall i = 1 \dots m, \forall j = 1 \dots n \quad (1.6) \\ \sum_{i=1}^m \mathbf{x}_{ij} \geq 1 \quad \forall j = 1 \dots n \quad (1.7) \\ \mathbf{x}_{ij}, \mathbf{y}_i \in \{0; 1\} \end{array} \right.$$

avec les notations :

- \mathbf{x}_{ij} vaut 1 si le client j est raccordé à l'équipement situé en i , 0 sinon ;
- \mathbf{y}_i vaut 1 si un équipement est placé en i , 0 sinon ;
- \mathbf{X} est la matrice des \mathbf{x}_{ij} ;
- f_i représente le coût d'installation d'un équipement en i . Dans le problème correspondant au Weber multi-source avec nombre d'équipements inconnus, on a : $f_i = C \forall i$;
- c_{ij} représente le coût de raccordement du client j à l'équipement en i ;

La contrainte (1.7) signifie que chaque client est raccordé à au moins un équipement, tandis que la contrainte (1.6) empêche un client d'être rattaché à un site où aucun équipement n'est installé. Comme il n'y a pas de contrainte de capacité sur les équipements, chaque client se raccordera à l'équipement le plus proche.

La résolution de (LS) nous fournira non seulement un raccordement initial des clients, mais aussi un nombre d'équipements et une borne supérieure de la fonction de coût pour le problème de Weber multi-source correspondant. Bien entendu, on espère obtenir précisément le nombre d'équipements et le raccordement optimaux, mais ce n'est pas forcément le cas. En effet, même si les caractéristiques des solutions des problèmes discret et continu présentent des similitudes, elles ne sont pas équivalentes pour autant.

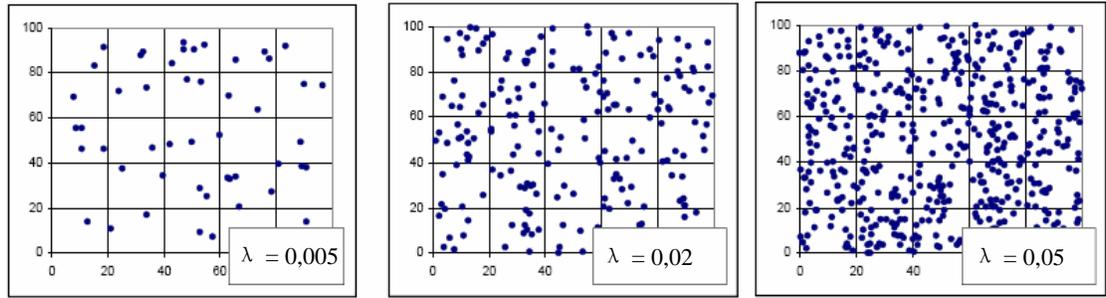


FIGURE 4.1 – Exemples de réalisations de processus ponctuels homogène de Poisson pour différentes valeurs de λ

4.1.2 Géométrie Stochastique

Le fait de manipuler des instances réelles de problèmes de localisation de grande taille implique d'avoir à sa disposition l'ensemble des données. Or ces données ne sont pas faciles à obtenir dans leur intégralité. L'idée principale d'une approche statistique consiste à écarter la difficulté due au très grand nombre de données en élaborant un modèle représentant la réalité sans avoir nécessairement l'ensemble des données. Pour cela, l'emplacement des clients, et des équipements sont modélisés à l'aide d'outils de géométrie stochastique nommés processus ponctuels spatiaux. Ces processus aléatoires sont caractérisés par leur loi de probabilité qui détermine de quelle manière sont répartis les points dans l'espace. Dans le cas qui nous intéresse ici, chaque niveau est modélisé par un processus ponctuel homogène de Poisson. Comme leur nom l'indique, les points sont répartis de manière homogène dans le plan et l'unique paramètre de leur loi est l'intensité λ qui correspond au nombre moyen d'éléments par unité de surface. Cette représentation est donc une simplification importante de la réalité. La figure 4.1 permet de constater l'influence de l'intensité λ sur la répartition des points.

Le but de l'approche statistique est donc de caractériser le processus qui modélise les équipements (i.e. trouver le paramètre λ correspondant) de façon à minimiser les coûts en moyenne.

Définition 4.1. *Dans un plan, étant donné un ensemble E de points, on appelle cellule de Voronoï centrée sur un point p appartenant à E , l'ensemble des points du plan qui sont plus proches de p que d'un autre point de E .*

Le fait de relier les clients à l'équipement le plus proche forme donc un pavage de cellules de Voronoï centrées chacune sur un équipement. Les clients se trouvant dans la cellule de Voronoï centrée sur un équipement seront reliés à cet équipement.

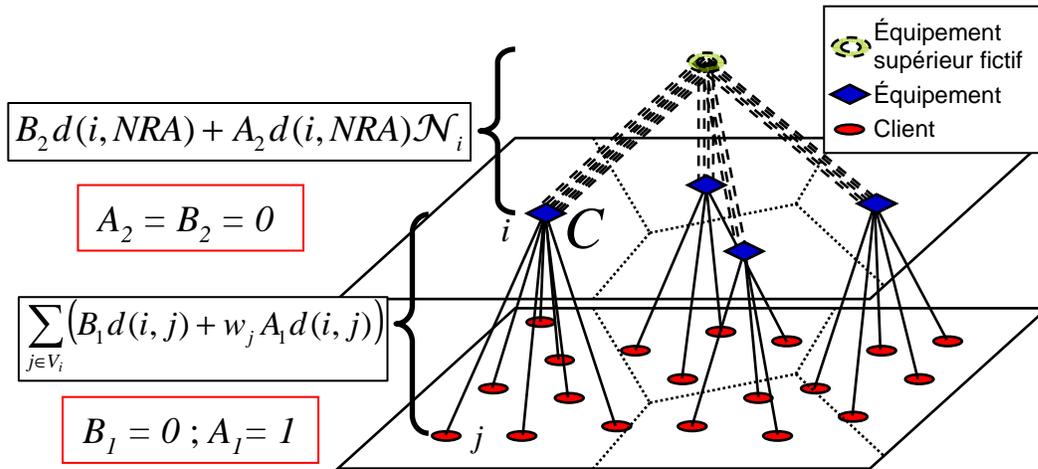


FIGURE 4.2 – Adaptation du modèle de réseau hiérarchique à 3 niveaux pour revenir à un problème de Weber multi-source

Cette approche statistique est étudiée dans [1] pour un problème de conception réseau hiérarchique à trois niveaux. Elle permet d'obtenir le paramètre, noté λ_1^* , du processus représentant les équipements du niveau intermédiaire, qui minimise l'espérance de la fonction de coût. Ce résultat est obtenu à l'aide des paramètres des processus représentant les autres niveaux, ainsi que les paramètres de la fonction de coût.

En considérant un niveau fictif d'équipements supérieurs et en adaptant les paramètres de coûts, cette méthode peut être adaptée au problème de Weber multi-source, comme illustré dans la figure 4.2. Pour cela, il nous faut calculer les paramètres λ_2 et λ_0 correspondant aux intensités respectives des processus représentant les clients d'une part, et l'équipement supérieur fictif d'autre part. Pour obtenir λ_2 , nous calculons la densité de la demande en divisant le nombre de clients par la surface de l'instance. Pour λ_0 , nous considérons qu'un seul équipement supérieur fictif est présent sur la zone. La densité de ce niveau fictif d'équipement est donc l'inverse de la surface de l'instance. La surface de l'instance peut être calculée comme étant la surface de l'enveloppe convexe des clients.

Cette méthode fait abstraction de la taille du problème. Les autres avantages sont le fait de pouvoir prendre en compte des modèles de coût complexes et le fait que les calculs pour obtenir λ_1^* se font rapidement, puisqu'il est obtenu directement par formule. De plus, les algorithmes pour obtenir des réalisations des processus des différents niveaux sont connus et efficaces. Cela permet d'obtenir un grand nombre de solutions initiales générées à l'aide du processus ponctuel représentant les équipements intermédiaires. Le nombre d'équipements présents dans ces solutions sert

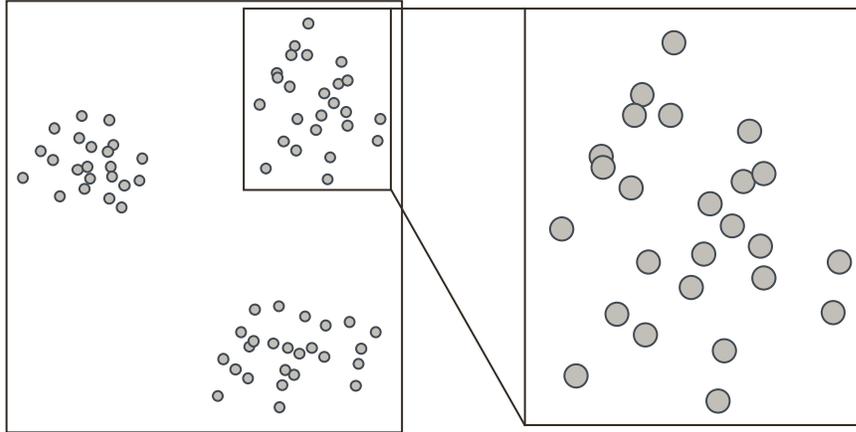


FIGURE 4.3 – Différence dans l'homogénéité de la distribution des clients selon l'échelle

alors d'estimation du nombre d'équipements optimal.

Néanmoins, ce λ_1^* correspond en fait à un nombre moyen d'équipements par unité de surface. Les résultats de cette méthode restent donc uniquement des résultats statistiques. Ils ne permettent pas d'obtenir une solution précise pour une instance donnée, au contraire d'une approche qui serait basée sur la recherche opérationnelle. De plus, le modèle utilisé dans [1] suppose que la distribution des clients est homogène et ne suit pas de configuration particulière. Ceci est loin d'être le cas de toutes les instances. Sur des instances réelles, cela peut notamment dépendre de l'échelle à laquelle on se place, comme l'illustre la figure 4.3.

4.2 Heuristiques basées sur LOCALLOC

Une heuristique connue pour les problèmes de localisation-allocation est celle de [10] qui s'appuie sur une décomposition du problème en localisation d'une part, et en allocation d'une autre. Pour une allocation donnée des clients aux équipements, il est facile de localiser les équipements de façon optimale. De même, pour une localisation donnée des équipements, il est facile de raccorder chaque client à l'équipement le plus proche. L'heuristique LOCALLOC alterne donc entre phases d'affectation des clients aux équipements, aussi nommées phases d'allocation, et phases de localisation de ces équipements dans le plan.

Phase de localisation : Pour une solution réalisable donnée $(\hat{X}, \hat{r}, \hat{s})$, chaque équipement i sera localisé de façon optimale vis-à-vis des clients qui lui sont affectés. Pour cela, il suffit de résoudre un problème de Weber simple pour

chaque équipement en ne considérant que ses clients.

Phase d'allocation : À l'issue d'une phase de localisation, il est possible que certains clients ne soient plus affectés à l'équipement le plus proche d'eux. Leur affectation est alors changée en conséquence. Une fois l'affectation de tous les clients vérifiée, une nouvelle phase de localisation a lieu si des changements ont été faits, sinon l'algorithme stoppe.

Le problème de cette heuristique vient de la qualité de l'optimum local trouvé qui dépend pour beaucoup de la ou des solutions initiales employées dans l'heuristique. Ainsi, dans [16], sur un problème de Weber multi-source classique avec 50 clients et 5 équipements, 200 solutions initiales aléatoires peuvent mener à 61 optimums locaux avec un écart relatif d'environ 40% entre le coût minimum et le coût maximum des solutions trouvées. Il est donc important d'obtenir une ou plusieurs bonnes solutions de départ.

L'utilisation de LOCALLOC avec les deux différentes méthodes décrites dans la section 4.1 pour trouver une estimation du nombre d'équipements optimal produit deux heuristiques différentes de résolution du problème de Weber multi-source avec nombre d'équipements inconnu :

- L'heuristique multi-phase de [7] propose de résoudre le problème de localisation simple associé, puis d'utiliser LOCALLOC sur la solution obtenue. Dans cet article, les auteurs proposent également une troisième phase optionnelle, qui consisterait à résoudre des problèmes de Weber classiques pour un nombre d'équipements proche de celui obtenu lors de la résolution du problème de localisation simple.
- Notre heuristique hybride, développée initialement pour l'application du chapitre 5, utilise l'approche de géométrie stochastique présentée dans 4.1.2 pour obtenir une estimation de la densité optimale d'équipements. Une fois cette densité obtenue, un nombre donné de solutions initiales sont générées comme réalisations du processus ponctuel de Poisson modélisant les équipements. LOCALLOC est appliquée à chacune de ces solutions initiales et nous conservons la meilleure solution.

4.3 Une méthode de Recuit simulé

De nombreuses méta-heuristiques ont été développées pour résoudre le problème de Weber multi-source classique. Pour une étude plus approfondie sur ces différentes méthodes, voir [6]. Dans ces méthodes, le principe est d'explorer l'espace des solutions admissibles en effectuant des mouvements dans certains voisinages de la solution courante. Le but est de converger petit à petit vers la meilleure solution dans une région précise, tout en évitant de s'enfermer trop vite dans un optimum local. Pour adapter ces méta-heuristiques au cas où le nombre d'équipements m à placer est inconnu, il faut que les voisinages considérés permettent de faire varier ce nombre d'équipements.

Nous proposons ici une méthode de recuit simulé pour résoudre le problème de Weber multi-source avec nombre d'équipements inconnu. Rappelons le principe des méthodes de recuit simulé :

- Au départ, plusieurs paramètres sont fixés :
 - la température initiale T_0 ;
 - le nombre d'itérations K par palier de température ;
 - le facteur de diminution X de la température à chaque palier ;
 - les différents critères d'arrêt possibles : nombre maximum de paliers, temps de résolution maximum, taux minimum de mouvements améliorant la solution au cours d'un palier.
 - la solution initiale.
- À chaque itération i d'un palier k de température, une solution de valeur c_{new} est choisie dans le voisinage de la solution courante (de valeur c_{cur}). Si cette solution est meilleure, elle devient la solution courante. Sinon, elle a une probabilité $e^{-\frac{c_{new}-c_{cur}}{T_k}}$ d'être acceptée comme nouvelle solution courante.
- Après K itérations au palier k , la température est réduite : $T_{k+1} \leftarrow X T_k$.
- L'algorithme stoppe lorsque l'un des critères d'arrêt est atteint.

Au départ de l'algorithme, la température est élevée. La probabilité d'accepter un mouvement qui dégrade la solution est donc haute, et les mouvements dans l'espace des solutions se font de façon presque aléatoire. Au fur et à mesure, les mouvements qui n'améliorent pas la solution sont de plus en plus rejetés et l'algorithme finit par converger vers un minimum local.

Dans notre méthode, une solution à notre problème sera décrite par l'ensemble des groupes de clients qui sont rattachés au même équipement. On effectue donc une partition de l'ensemble des clients, le nombre de classes représentant alors le nombre d'équipements. Les mouvements pratiqués sur la solution consisteront à transférer un client d'un équipement à un autre. Il s'agit d'un voisinage classique dans les méta-heuristiques utilisées sur des problèmes de localisation-allocation. Pour faire évoluer le nombre de classes, à chaque transfert de client il est possible que celui-ci soit rattaché à un nouvel équipement qui formera une nouvelle classe. Parallèlement, si une classe se voit enlever son dernier client, elle est supprimée et l'équipement correspondant disparaît avec elle. L'utilisation d'autres voisinage relocalisant les équipements comme ceux présentés dans [6] pourraient aussi être envisagés ici.

Afin d'obtenir des ensembles de clients dont l'enveloppe convexe est disjointe (i.e pas de client isolé au milieu d'un groupe qui n'est pas le sien) au départ ou à chaque palier de température, on peut utiliser une itération de LOCALLOC. Pour les instances de grande taille, l'espace des solutions est très vaste. Par conséquent le nombre de mouvements nécessaires pour affiner une solution est très élevé. Pour accélérer la convergence de l'algorithme il est possible d'utiliser plus activement LOCALLOC lors des différents paliers de température. Différentes variantes sont possibles selon

le nombre d'itérations de LOCALLOC que l'on effectue à chaque palier. Une première variante peut procéder à deux itérations de LOCALLOC, c'est dire que par deux fois les clients sont raccordés à l'équipement le plus proche et les équipements affectés par ces changements relocalisés. Une variante pourrait consister à laisser LOCALLOC atteindre un minimum local à chaque palier. Lors de ces utilisations de LOCALLOC, si un équipement se retrouve sans client, il est supprimé.

Le recuit simulé peut à la fois être utilisé comme méthode de résolution à part entière, ou intervenir en post-optimisation à partir de solutions obtenues par d'autres méthodes.

4.4 Méthodes de générations de colonnes

Plusieurs méthodes de génération de colonnes ont été proposées pour le problème de Weber multi-source classique (par exemple [40], [23], et plus récemment [39]). Ces méthodes modélisent le problème de Weber multi-source comme un problème de couverture d'ensemble.

Tout comme pour la première phase de l'heuristique multi-phase, l'idée est de discrétiser le problème pour se débarrasser de la distance euclidienne. En apparence, nous ne disposons pas d'emplacements potentiels pour les équipements. Mais dans la solution optimale, les équipements ne peuvent pas être placés n'importe où dans le plan. Ils sont placés de façon optimale par rapport à leurs clients. Un seul emplacement est donc possible par groupe de clients. Si n est le nombre de clients, le nombre de groupes de clients possibles à l'exception du groupe vide est $2^n - 1$. Nous allons donc résoudre le problème non plus en localisant les équipements et en leur affectant des clients, mais en décidant quels groupes de clients se verront affecter le même équipement.

Chaque client doit alors être couvert par un groupe de clients. L'équipement de chaque groupe est localisé de façon optimale par rapport au clients de ce groupe. Les différents groupes de clients possibles forment les variables du problème. Le problème

de Weber multi-source classique est donc modélisé de la façon suivante :

$$(\mathcal{WM}_{setcover}) \left\{ \begin{array}{l} \min_{\mathbf{z}} \sum_{k=1}^{(2^n-1)} c^k \mathbf{z}_k \\ \text{s.c.} \\ \sum_{k=1}^{(2^n-1)} x_j^k \mathbf{z}_k \geq 1 \quad \forall j = 1 \dots n \quad (4.1) \\ \sum_{k=1}^{(2^n-1)} \mathbf{z}_k = p \quad (4.2) \\ \mathbf{z}_k \in \{0; 1\} \quad \forall k \end{array} \right.$$

- \mathbf{z}_k vaut 1 si le groupe de clients k est utilisé, 0 sinon ;
- x_j^k vaut 1 si le groupe de clients k contient le client j , 0 sinon ;
- $c^k = \min_{\mathbf{r}, \mathbf{s}} \sum_{j=1}^n x_j^k w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2}$, correspond au coût total du groupe k :
équipement plus liaisons avec les clients.

Le nombre de variables de ce problème étant exponentiel, il est résolu à l'aide de méthodes de générations de colonnes. Le coût réduit \bar{c}^k de la colonne k est égal à :

$$\begin{aligned} \bar{c}^k &= c^k - \sum_{j=1}^n x_j^k u_j + \mu \\ &= \min_{\mathbf{r}, \mathbf{s}} \sum_{j=1}^n x_j^k w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} - \sum_{j=1}^n x_j^k u_j + \mu \\ &= \min_{\mathbf{r}, \mathbf{s}} \sum_{j=1}^n x_j^k \left(w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} - u_j \right) + \mu \end{aligned}$$

où les u_j sont les variables duales associées aux contraintes (4.1), et μ la variable duale associée à la contrainte (4.2). Le sous-problème, consistant à trouver la colonne de coût réduit minimum, se présente donc sous la forme suivante :

$$\min_{\mathbf{x}, \mathbf{r}, \mathbf{s}} \sum_{j=1}^n \mathbf{x}_j \left(w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} - u_j \right) \quad (4.3)$$

Dans le problème (4.3) la variable duale μ n'intervient pas. Or le problème de Weber multi-source avec nombre d'équipements inconnus se modélise sous la forme d'un

problème de couverture d'ensemble de la façon suivante :

$$(\mathcal{WM}_{\text{équipements}_{\text{setcover}}}) \left\{ \begin{array}{l} \min_{\mathbf{z}} \sum_{k=1}^{(2^n-1)} c^k \mathbf{z}_k \\ \text{s.c.} \\ \sum_{k=1}^{(2^n-1)} x_j^k \mathbf{z}_k \geq 1 \quad \forall j = 1 \dots n \\ \mathbf{z}_k \in \{0; 1\} \quad \forall k \end{array} \right. \quad (4.1)$$

Par rapport au problème $(\mathcal{WM}_{\text{setcover}})$, la contrainte (4.2) a été enlevée et le coût d'un groupe prend en compte le coût d'installation de l'équipement :

$$c^k = C + \min_{\mathbf{r}, \mathbf{s}} \sum_{j=1}^n x_j^k w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2}$$

Le coût réduit de la colonne k devient :

$$\begin{aligned} \bar{c}^k &= c^k - \sum_{j=1}^n x_j^k u_j \\ &= C + \min_{\mathbf{r}, \mathbf{s}} \sum_{j=1}^n x_j^k w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} - \sum_{j=1}^n x_j^k u_j \\ &= C + \min_{\mathbf{r}, \mathbf{s}} \sum_{j=1}^n x_j^k \left(w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} - u_j \right) \end{aligned}$$

et le sous-problème est identique à celui de $(\mathcal{WM}_{\text{setcover}})$: il s'agit de (4.3).

Les méthodes de générations de colonnes développées pour le Weber multi-source classique peuvent donc être adaptées très simplement au problème de Weber multi-source avec nombre d'équipements inconnus.

Le sous-problème (4.3) correspond à un problème de Weber avec distances maximales. En effet, un client j apportera une contribution négative au coût réduit si et seulement si :

$$\begin{aligned} w_j \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} - u_j &\leq 0 \\ \Leftrightarrow \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} &\leq \frac{u_j}{w_j} \end{aligned} \quad (4.4)$$

La quantité $\frac{u_j}{w_j}$ agit donc comme une distance maximum entre l'équipement et le client j au-delà de laquelle le client n'est pas rattaché à l'équipement, comme illustré dans la figure 4.4.

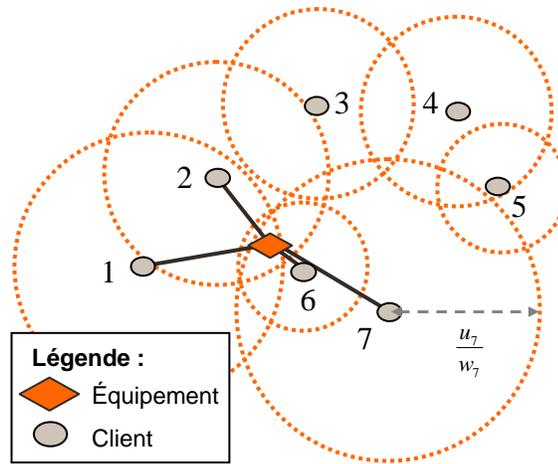


FIGURE 4.4 – Exemple de problème de Weber avec distances maximales

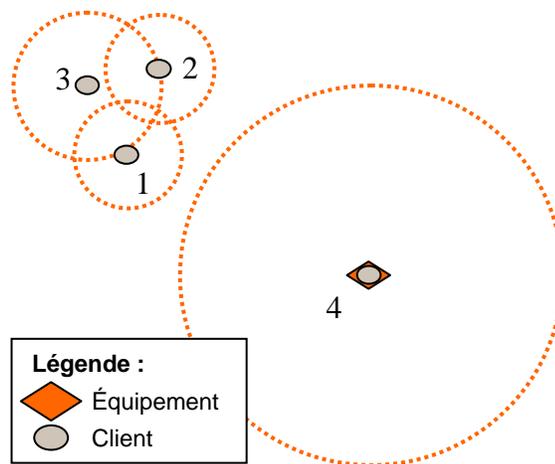


FIGURE 4.5 – Problème de Weber avec distances maximales où la solution est d'installer l'équipement sur le client isolé 4

Dans [23], l'auteur propose de résoudre ce problème comme un programme D. C. (différence de fonction convexe). Dans [39] les auteurs utilisent l'algorithme de résolution du problème de Weber avec distances maximales développé dans [15]. Les auteurs de [15] montrent qu'il est possible de résoudre le problème de Weber avec distances maximales en résolvant un nombre polynomial de problèmes de Weber simples. Ils montrent que les cercles centrés sur les clients et de rayon la distance maximale pour ce client, partitionnent l'espace au maximum en $2n(n-1)$ régions, ainsi que la propriété suivante :

Propriété 4.1 (voir Lemme 1 dans [15]). *Il existe une solution optimale (\mathbf{x}^*, r^*, s^*) du sous-problème telle que :*

$$\forall j = 1 \dots n, \quad x_j^* = 1 \Leftrightarrow \sqrt{(r^* - a_j)^2 + (s^* - b_j)^2} \leq \frac{u_j}{w_j} \quad (4.5)$$

La propriété 4.1 signifie que les clients qui sont rattachés à l'équipement dans la solution optimale sont *tous* ceux vérifiant l'inégalité 4.4. Par exemple, dans le cas de la figure 4.4, il s'agit des clients 1, 2, 6, et 7. Chaque région correspond donc à un et un seul groupe de clients potentiellement optimal. [15] propose donc de résoudre le problème en énumérant les régions à partir des points d'intersection des cercles, puis en résolvant un problème de Weber simple pour chacun des groupes de clients ainsi obtenus.

L'algorithme utilisé dans [39] pour énumérer les groupes est le suivant, où $d(i, j)$ représente la distance euclidienne entre le client i et le client j :

Algorithme 4.1.

Initialisation de $\mathcal{G} = \emptyset$, l'ensemble des clients.

```

for  $i = 1 \dots n$  do
  for  $j = i + 1 \dots n$  do
    if  $d(i, j) \leq \frac{u_i}{w_i} + \frac{u_j}{w_j}$  then {si les deux cercles s'intersectent ou que l'un est
      contenu dans l'autre}
      Trouver l'ensemble  $\mathcal{I}^{ij}$  des points d'intersections des cercles des clients  $i$  et
       $j$ ;
      for all  $t \in \mathcal{I}^{ij}$  do
         $G := \emptyset$ ;
        for all  $k = 1 \dots n \ \&\& \ k \neq i \ \&\& \ k \neq j$  do
          if  $t$  est à l'intérieur du cercle centré sur  $k$  then
             $G := G \cup \{k\}$ ;
          end if
        end for
         $\mathcal{G} := \mathcal{G} \cup \{G\} \cup \{G \cup \{i\}\} \cup \{G \cup \{j\}\} \cup \{G \cup \{i, j\}\}$  {On rajoute à
          l'ensemble des groupes les 4 groupes définis par le point d'intersection  $t$ }
        end for
      end if
    end for
  end for

```

Cet algorithme présente des défauts lorsque certains cercles n'intersectent aucun autre, par exemple lorsqu'un client est très éloigné des autres, ou lorsque deux clients sont proches et que l'un des cercles est contenu dans l'autre. Dans ces cas là, les régions du plan correspondantes ne sont pas définies par les points d'intersection des cercles. Elles ne sont donc pas prises en compte dans l'algorithme de [39] (ni de [15]). L'équipement optimal peut pourtant se trouver dans ces zones, par exemple dans le problème de la figure 4.5.

Pour tenir compte des cercles isolés ou entièrement contenus dans d'autres cercles sans présenter aucun point d'intersection, nous proposons les modifications suivantes dans l'algorithme d'énumération des groupes de clients potentiellement optimaux :

Algorithme 4.2.

Initialisation de $\mathcal{G} = \emptyset$, l'ensemble des clients.

```

for  $i = 1 \dots n$  do
  Initialisation de  $C_i := \emptyset$ , l'ensemble des clients dont le cercle contient le cercle
  centré en  $i$ .
  end for
for  $i = 1 \dots n$  do
   $flag = 0$  {initialisation du compteur des points d'intersection du cercle  $i$  avec
  les autres cercles}
  for  $j = i + 1 \dots n$  do

```

```

if  $d(i, j) \leq \frac{u_i}{w_i} + \frac{u_j}{w_j}$  then {si les deux cercles s'intersectent ou que l'un est
contenu dans l'autre}
  if  $d(i, j) + \frac{u_i}{w_i} < \frac{u_j}{w_j}$  then {si le cercle centré en  $i$  est contenu dans celui
centré en  $j$ }
     $C_i := C_i \cup \{j\}$ ;
  else if  $d(i, j) + \frac{u_j}{w_j} < \frac{u_i}{w_i}$  then {si le cercle centré en  $j$  est contenu dans
celui centré en  $i$ }
     $C_j := C_j \cup \{i\}$ ;
  else
     $flag++$ ;
    Trouver l'ensemble  $\mathcal{I}^{ij}$  des points d'intersections des cercles des clients  $i$ 
et  $j$ ;
    for all  $t \in \mathcal{I}^{ij}$  do
       $G := \emptyset$ ;
      for all  $k = 1 \dots n \ \&\& \ k \neq i \ \&\& \ k \neq j$  do
        if  $t$  est à l'intérieur du cercle centré sur  $k$  then
           $G := G \cup \{k\}$ ;
        end if
      end for
       $\mathcal{G} := \mathcal{G} \cup \{G\} \cup \{G \cup \{i\}\} \cup \{G \cup \{j\}\} \cup \{G \cup \{i, j\}\}$  {On rajoute à
l'ensemble des groupes les 4 groupes définis par le point d'intersection
 $t$ }
    end for
  end if
end if
end for
if  $flag == 0$  then {si le cercle centré en  $i$  n'intersecte pas les autres}
   $\mathcal{G} := \mathcal{G} \cup \{C_i \cup \{i\}\}$ ;
end if
end for

```

Une fois l'ensemble des groupes de clients potentiellement optimaux obtenus, un problème de Weber simple est résolu pour chacun d'entre eux afin de déterminer la solution optimale. [15] et [39] proposent de faire appel à des bornes inférieures pour réduire le nombre de groupes de clients à évaluer. Dans les instances très denses, le nombre de solutions à évaluer peut être très élevé, et la résolution du problème de Weber avec distances maximum peut donc prendre beaucoup de temps malgré le fait que l'algorithme soit polynomial en n et en le temps nécessaire à la résolution d'un Weber simple. Toutefois, dans le cadre de la résolution du sous-problème associé au Weber multi-source, un avantage de cet algorithme d'énumération est que l'on obtient un grand nombre de colonnes de coût réduit négatif, que l'on est libre d'ajouter au

problème maître en plus de la colonne optimale.

Pour résoudre le problème en nombre entiers, il faut implémenter un algorithme de branch & price. Nous n'avons pas implémenté nous même de branch & price. La génération de colonnes n'étant pas assez efficace pour résoudre les tailles d'instance rencontrées dans l'application du chapitre (5), l'investissement nécessaire pour implémenter un algorithme de branch & price était trop lourd en regard des résultats escomptés.

Dans [39], les auteurs implémentent un algorithme de branch & price pour résoudre le problème de Weber multi-source classique. Les décisions de branchement sont effectuées selon la règle de Ryan & Foster [42] (voir section (1.6)). À un noeud donné, chaque client a donc un ensemble de clients qui doivent être couverts par la même colonne que lui, et un ensemble de clients qui doivent être couverts par des colonnes différentes. Les auteurs de [39] proposent d'intégrer ces règles de branchements de manière implicite au sous-problème en écartant tout groupe de clients ne correspondant aux décisions de branchement parmi ceux générés lors de la résolution du sous-problème. Il faut faire extrêmement attention à la façon dont ceci est réalisé. Les auteurs de [39] expliquent qu'ils éliminent les mauvaises colonnes dès leur génération. Dans ce cas, il n'est plus garanti que l'algorithme de résolution du sous-problème trouve la colonne de coût réduit minimal. En effet, cet algorithme n'énumère pas toute les colonnes de coût réduit négatif, seulement celles étant potentiellement optimales pour le sous-problème sans décision de branchement. Or, la colonne optimale du problème avec décisions de branchements ne fait pas forcément partie de celles-ci. Prenons l'exemple de la figure (4.6). Supposons que les décisions de branchements imposent que le client 4 soit couvert par une colonne ne couvrant ni le client 1, ni les clients 2 et 3. La colonne optimale correspond alors à l'ensemble de clients $\{1, 2, 3\}$ dont l'équipement est placé dans la zone hachurée. Mais cette colonne ne sera jamais énumérée par l'algorithme de résolution du sous-problème. En effet ce dernier construira pour la zone hachurée la colonne $\{1, 2, 3, 4\}$, qui sera éliminée car ne respectant pas les décisions de branchement.

Ce problème vient du fait que la propriété 4.1 n'est plus valide lorsqu'on est en présence de règles de branchement empêchant certains clients d'être dans la même colonne. Le meilleur groupe de clients pour chaque région est alors plus difficile à trouver. Ce nouveau sous-problème reste à étudier.

4.5 Comparaison des différentes méthodes

Dans cette section nous présentons les résultats numériques obtenus avec les différentes méthodes présentées dans ce chapitre sur deux instances issues de TSPLIB ([38]) et utilisées notamment dans [7] et dans [39]. La première est l'instance p654 qui

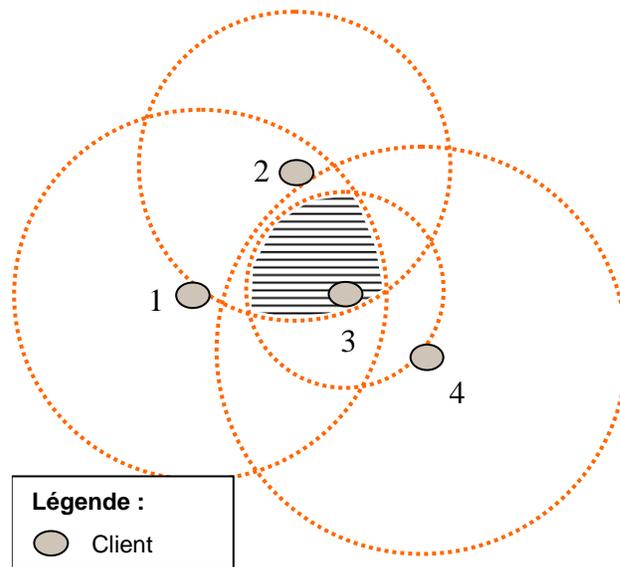


FIGURE 4.6 – Exemple de cas où les décisions de branchement peuvent empêcher de trouver la solution optimale du sous-problème

comporte 654 clients principalement répartis sur quatre zones. La seconde est l'instance u1060 qui comporte 1060 clients et où ces derniers sont répartis de façon plus homogène. La figure 4.7 présente la répartition des clients dans les deux instances. Deux valeurs différentes par instance du coût d'installation C d'un équipement ont été utilisées.

Heuristique multiphase : Dans notre implémentation, le problème de localisation simple associé est résolu de façon exacte par un algorithme de branch & bound s'appuyant sur la méthode primal-duale de [22]. Ceci explique les légères différences de nos solutions avec celles de [7]. Nous n'avons pas utilisé la troisième phase optionnelle faisant varier le nombre de d'équipements.

Heuristique avec géométrie stochastique : Dans notre implémentation, 100 tirages sont effectués avec l'estimation de densité optimale. LOCALLOC est utilisée sur tous les tirages et nous avons gardé le meilleur en termes de coût.

Recuit simulé : Le recuit simulé est ici utilisé comme méthode de résolution à part entière. Le nombre d'équipement est initialisé à $n/10$, et l'affectation initiale de chaque client est tirée au hasard. LOCALLOC n'est pas utilisé au cours du recuit. Les résultats présentés sont des moyennes obtenues sur 50 exécutions de l'algorithme.

Génération de colonnes : Seule la relaxation continue est ici résolue pour obtenir une borne inférieure. Dans certains cas, la solution de la relaxation continue

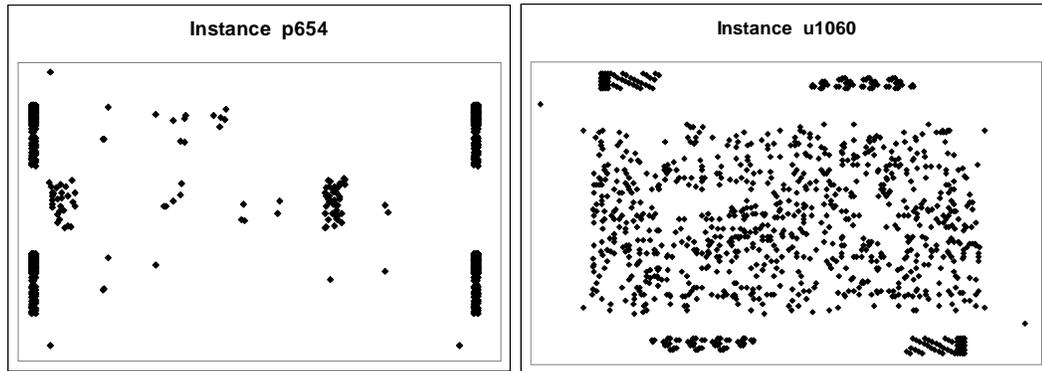


FIGURE 4.7 – Topologies des instances p654 et u1060

est entière. Nous avons utilisées les deux méthodes de génération de colonnes présentées dans les chapitres 1 et 2, et comparées dans la section 2.5. Les deux générations de colonnes ont été initialisées avec la solution de l’heuristique multi-phase et 50 tirages de l’heuristique avec géométrie stochastique.

Les tests des heuristiques et du recuit simulé ont été réalisés sur un serveur IBM XSERIES_455 disposant de quatre processeurs cadencés à 3Ghz (un seul est utilisé) et de 10Go de mémoire vive. Les tests des générations de colonnes ont quant eux été réalisés sur un serveur linux équipé d’un processeur xeon 3,6Ghz et de 4Go de mémoire vive.

Les résultats obtenus sont présentés dans le tableau 4.1. Pour la génération de colonnes, sont indiqués : la valeur de la solution de la relaxation continue ; le temps CPU utilisé par les deux méthodes ; si oui ou non la solution de la relaxation continue est entière ; et si elle l’est, le nombre m^* d’équipements présents dans la solution optimale. Pour chaque heuristique sont indiqués : la valeur de la solution obtenue ; l’écart relatif avec la borne obtenue par génération de colonnes ; le temps CPU utilisé ; et le nombre m d’équipements dans la solution ainsi que la différence avec m^* .

Premier constat, nous avons ici la confirmation que l’heuristique multi-phase de [7] fournit des solutions d’excellente qualité avec au maximum 0.15% d’écart avec le coût optimal et un nombre d’équipements dans les solutions quasi identique à celui des solutions optimales, et ceci avec un temps CPU utilisé très réduit (quelques secondes). À propos du temps CPU utilisé par cette méthode, il est intéressant de noter que dans nos tests, le temps utilisé par l’heuristique multi-phase augmente lorsque le ratio m/n diminue (ceci est confirmé par les tests présentés au chapitre 5). L’inverse est observé dans les tests présentés dans [7]. L’heuristique de résolution du problème de localisation simple qu’il utilisent semble donc plus adaptée à des ratios m/n faibles, tandis que la méthode de [22] semble plus adaptées à des ratios m/n

	Instance	p654		u1060	
	C	500	1000	2000	8000
Heuristique multi-phase	Coût	54191.178	73854.237	483003.164	849002.311
	Écart à l'optimum	0.03%	< 0.01%	0.15%	0.02%
	Temps CPU (s)	4	6	2	24
	$m (m - m^*)$	57 (+1)	32 (-1)	95 (0)	42 (?)
Heuristique Géostoch	Coût	60295.056	81845.075	498012.041	869385.647
	Écart à l'optimum	11.3%	10.8%	3.26%	2.42%
	Temps CPU (s)	48	36	117	88
	$m (m - m^*)$	49 (-7)	33 (0)	105 (+10)	44 (?)
Recuit simulé	Coût	57210.094	82009.789	503410.291	935280.288
	Écart à l'optimum	5.6%	11.1%	4.38%	10.2%
	Temps CPU (s)	214.42	207.56	377.14	344.12
	$m (m - m^*)$	61.74 (+5.74)	43.64 (+10.64)	94.26 (-0.74)	68.4 (?)
Génération de colonnes	Coût	54175.193	73847.856	482282.62	848862.48
	Temps CPU classique (s)	3496	2529	4394	41935
	Temps CPU centrale (s)	2063	2220	1210	5776
	Solution entière	oui	oui	oui	non
	m^*	56	33	95	?

TABLE 4.1 – Résultats des différentes méthodes sur les instances p654 et u1060

plus élevés.

Concernant, l'heuristique de géométrie stochastique, nous constatons sans surprise qu'elle produit de meilleures solutions sur l'instance u1060, où la répartition des clients est la plus homogène. Mais elle fait moins bien que l'heuristique multi-phase, tant au niveau de la qualité des solutions qu'au niveau du temps de calcul.

Les résultats du recuit simulé sont eux aussi de bien moins bonne qualité que ceux de l'heuristique multi-phase. On lui préférera également l'heuristique de géométrie stochastique qui fournit des solutions de qualité comparables en moins de temps.

On retrouve dans les résultats des deux générations de colonnes sur ces deux instances la stabilité supérieure de la génération de colonne centrale par rapport à la génération de colonne classique, qui dans le dernier test se perd dans les problèmes de dégénérescence. Les tests montrent également que la relaxation continue fournit souvent une solution entière. Les temps de calcul sont quant à eux très élevés et augmentent lorsque le ratio m/n diminue.

4.6 Conclusion

En conclusion, le développement de méthodes de génération de colonnes pour le problème de Weber multi-source avec nombre d'équipements inconnus aura permis de montrer l'excellente qualité de l'heuristique multi-phase présentée dans [7]. Les autres méthodes développées pour le problème du chapitre 5 obtiennent des performances bien moindre sur les deux instances de la littérature testées. Nous verrons toutefois dans le chapitre suivant que ces méthodes peuvent se montrer utiles lorsque les tailles d'instances augmentent, et que la résolution du problème de localisation dans la première phase de l'heuristique multi-phase devient plus difficile.

Chapitre 5

Déploiement de l'accès du Réseau Téléphonique Commuté sur un territoire vierge

Dans ce chapitre nous présentons un problème de localisation optimale d'équipements dans le réseau téléphonique commuté (RTC). Le problème se place dans la partie accès du réseau qui agrège le trafic des clients pour l'envoyer ensuite vers sa destination. Le réseau d'accès du RTC se décrit comme un réseau hiérarchique. Au niveau supérieur se trouve un concentrateur agrégeant le trafic d'un nombre de clients allant de plusieurs milliers à plusieurs dizaines de milliers, selon la densité de population de la zone considérée. Nous nous intéressons ici à la localisation optimale des équipements intermédiaires, entre ce concentrateur et les clients.

La taille des instances considérées est une difficulté majeure. Des hypothèses sont donc faites pour simplifier le problème. La localisation des équipements se fait dans le plan (sans emplacements potentiels prédéfinis), la distance prise en compte est la distance euclidienne et les liaisons se font à vol d'oiseau. On suppose également que le coût d'installation des équipements est le même quel que soit l'endroit où ils sont installés, et ils n'ont pas de capacité.

Une approche existante à France Télécom consiste à utiliser des outils de géométrie stochastique. Le but de ce chapitre est de proposer des approches de résolution basées sur des techniques d'optimisation combinatoire et de les comparer à l'approche existante. L'objectif est de pouvoir proposer un panel d'outils adaptés aux différentes difficultés et tailles d'instances.

Dans une première partie nous présenterons le problème et son contexte télécom. Nous présenterons ensuite les différentes modélisations du problème et les approches développées. Ces approches seront ensuite comparées sur des instances artificielles,

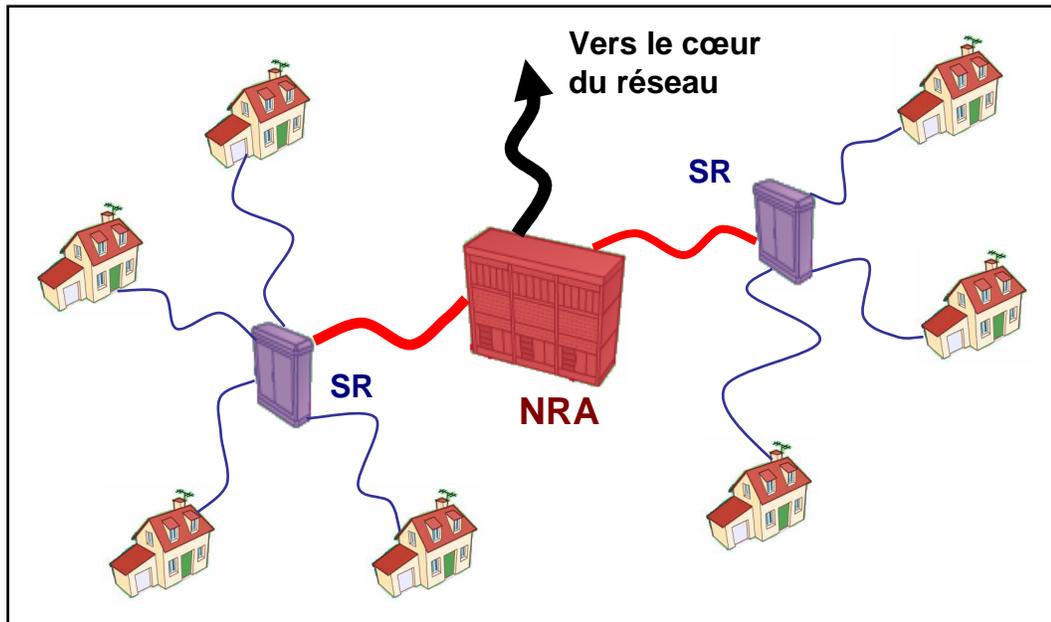


FIGURE 5.1 – Partie Accès du RTC pour une zone locale

ainsi que sur deux instances issues de données réelles.

5.1 Introduction

5.1.1 Architecture du Réseau Téléphonique Commuté

Le réseau téléphonique commuté est constitué de deux parties : le cœur du réseau et la partie accès. La partie accès relie les abonnés au cœur du réseau qui se charge d'acheminer les appels. La partie accès prend la forme d'un réseau arborescent à trois niveaux hiérarchiques (voir figure 5.1).

Au plus haut niveau se trouve les Nœuds de Raccordement des Abonnés (NRA), puis en dessous, au niveau intermédiaire, les Sous-Répartiteurs (SR), et enfin au plus bas niveau : les abonnés ou clients. En réalité, un dernier équipement, le point de concentration (PC), vient se placer après le SR et juste avant l'abonné. Par abus de langage et pour simplifier, nous assimilerons les PC à des clients. Ils correspondront à notre niveau hiérarchique le plus bas.

Une zone locale est une zone géographique dont les clients sont tous reliés au même NRA. La partie accès du RTC est donc constituée d'une multitude de zones locales, communiquant entre elles grâce au cœur du réseau. La figure 5.2 montre un exemple

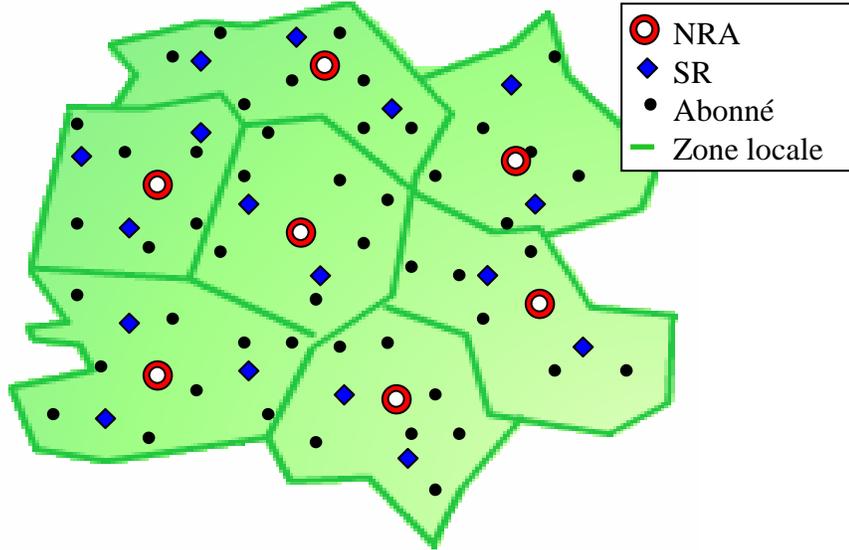


FIGURE 5.2 – Exemple de répartition des éléments du RTC dans le plan

de décomposition d'un territoire en zones locales.

5.1.2 Problème de localisation de sous-répartiteurs

Lors d'une migration technologique des SR, il est possible que le nouveau type d'équipement influe différemment sur les coûts. La localisation optimale de ces nouveaux équipements peut donc être différente de celle des anciens. Il faut alors déterminer le nombre et les emplacements des nouveaux équipements en tenant compte des éléments inchangés du réseau et de façon à minimiser les coûts. Pour l'implantation dans un nouveau territoire, les NRA sont fixés par des contraintes géographiques ou stratégiques et les emplacements des clients sont connus. Dans les deux cas, les emplacements des équipements des niveaux supérieur et inférieur (i.e. des NRA et des clients) sont fixés, soit historiquement dans le cas d'une migration technologique, soit par des contraintes géographiques, stratégiques ou autres dans le cas de l'implantation d'un nouveau réseau.

L'objectif est alors de minimiser la somme des coûts du réseau en trouvant le nombre et les emplacements optimaux pour les sous-répartiteurs. Nous nous plaçons pour cela à l'échelle d'une zone locale. Les coûts pris en considération sont les suivants :

Les coûts d'installation qui représentent les coûts d'achat et de mise en place des SR. Nous supposons qu'il s'agit du même coût constant C pour tous les SR.

Les coûts de transport qui représentent les coûts du raccordement des SR aux

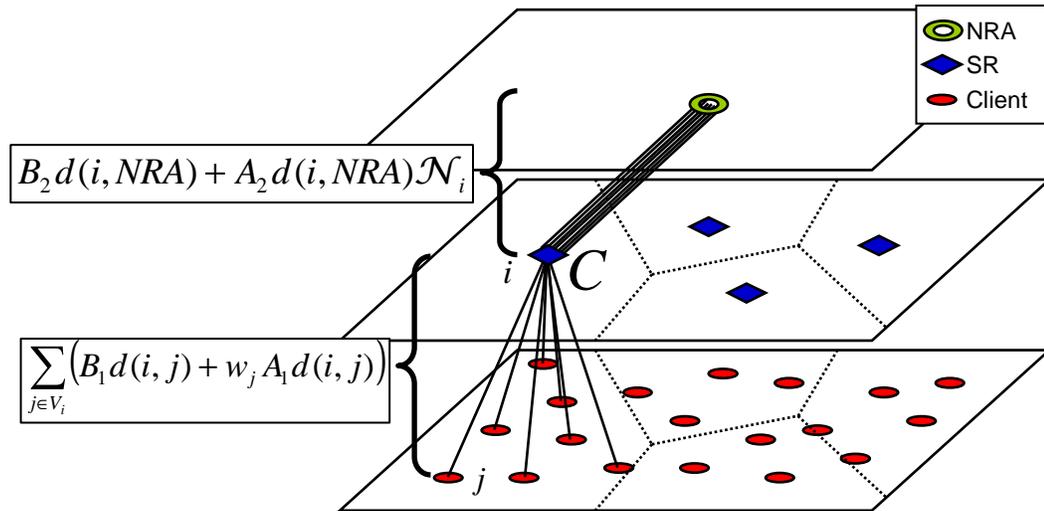


FIGURE 5.3 – Structure des coûts du réseau d'accès

éléments des deux autres niveaux du réseau. Les coûts de transport ont deux composantes :

- *Le coût d'infrastructure* qui est la somme des coûts des infrastructures nécessaires au raccordement. C'est une fonction de la distance entre les éléments raccordés dont les paramètres varient selon le type d'infrastructure (tranchées, poteaux, ...).
- *Le coût de capacité* qui est le coût du support physique (médium) utilisé pour le raccordement. C'est une fonction de la distance entre les éléments raccordés et de la demande transportée. Ses paramètres varient selon le type de médium (cuivre, fibre optique, ...).

Une des particularités de notre problème vient de la mutualisation des coût d'infrastructure au niveau de chaque liaison, contrairement aux coût de capacité. En pratique, cela signifie que l'on creusera une seule tranchée pour faire passer plusieurs câbles.

Pour résoudre notre problème, nous considérons que les liaisons entre les différents éléments se font à vol d'oiseau. L'organisation des coûts est représentée sur la figure 5.3, où :

- B_1 et B_2 sont respectivement les coûts d'infrastructure par mètre de liaison entre les clients et les SR, et entre les SR et le NRA.
- A_1 et A_2 sont respectivement les coûts de capacité par mètre de liaison et par unité de demande entre les clients et les SR, et entre les SR et le NRA.
- V_i est l'ensemble des clients rattachés au SR i .
- \mathcal{N}_i est la demande totale associée aux clients rattachés à i .

- w_j est la demande associée au client j .
- $d(i, j)$ et $d(i, NRA)$ sont respectivement la distance euclidienne entre le SR i et le client j , et entre le SR i et le NRA.

5.2 Modélisations

Dans cette section nous présenterons les deux principales modélisations de notre problème. La première est celle développée dans [1] en utilisant la géométrie stochastique. La deuxième est une modélisation sous forme de programme mathématique qui servira de base aux approches d'optimisation combinatoire.

5.2.1 Modélisation à l'aide de la géométrie stochastique

Dans [1], les auteurs proposent de modéliser des réseaux hiérarchiques à trois niveaux en utilisant des processus ponctuels pour modéliser chaque niveau d'équipements. C'est cette même approche que nous avons introduite dans la section 4.1.2.

Un processus ponctuel est un outil statistique décrivant la distribution de points dans l'espace à l'aide d'une loi de probabilité disposant d'un ou plusieurs paramètres. Les processus ponctuels utilisés dans [1] sont des processus ponctuels de Poisson homogènes. Dans ces processus, les points sont répartis de façon homogène dans le plan. Le seul paramètre du processus est son intensité λ correspondant à la densité de points (le nombre de points par unité de surface).

Les 3 niveaux du réseau d'accès sont chacun modélisés par un processus ponctuel de Poisson différent indépendant des autres. Notons λ_0 , λ_1 , et λ_2 les intensités respectives des processus des clients, des SR, et des NRA. Les auteurs de [1] expriment alors l'espérance de la fonction de coût du réseau rattaché à un NRA en fonction de ces trois paramètres et des structures de coûts présentées à la section 5.1.2 :

$$\mathcal{F}_{geostoch} = \left(\frac{B_2}{\sqrt{\pi}\lambda_2^{3/2}} \Gamma(1, 5) + \frac{C}{\lambda_2} \right) \lambda_1 + \left(\frac{A_2\lambda_0}{\sqrt{\pi}\lambda_2^{3/2}} \Gamma(1, 5) \right) + \left(\frac{(B_1 + A_1)\lambda_0}{\sqrt{\pi}\lambda_2} \Gamma(1, 5) \right) \lambda_1^{-\frac{1}{2}} \quad (5.1)$$

où Γ est la fonction gamma d'Euler.

Il est alors possible de calculer la valeur λ_1^* de l'intensité du processus des SR qui minimise l'espérance du coût :

$$\lambda_1^* = \lambda_2^{\frac{1}{3}} \left[\frac{\lambda_0 (A_1 + B_1)}{2B_2 + 4C\sqrt{\lambda_2}} \right]^{\frac{2}{3}} \quad (5.2)$$

Le grand avantage de cette approche est qu'elle fait totalement abstraction de la taille des données. Seuls quelques paramètres sont manipulés pour représenter la

réalité et le paramètre λ_1^* est obtenu directement par formule. À partir de là, il est possible de générer très efficacement un grand nombre de réalisations du processus ponctuel modélisant les SR. Chacune de ses réalisations peut être considérée comme une solution potentielle de notre problème de localisation.

Un inconvénient de cette méthode est qu'elle ne tient pas compte des détails de la configuration particulière de chaque instance. L'optimisation est faite *en moyenne* pour toutes les instances présentant une distribution de clients et de NRA similaires. De plus, cette approche fait l'hypothèse d'une répartition homogène des clients (et également des SR) en les modélisant par un processus ponctuel de Poisson et suppose que leur distribution est indépendante de celle des NRA. Or dans la réalité, ces hypothèses peuvent être fortement remises en cause.

Ces inconvénients ont motivé la recherche de méthodes de résolution prenant mieux en compte les spécificités de chaque instance. Pour cela nous avons utilisé l'optimisation combinatoire, même si celle-ci est beaucoup plus sensible à la taille des données à traiter.

5.2.2 Modélisation sous forme de programme mathématique

Nous voulons donc minimiser les coûts d'installation et de raccordement de notre réseau d'accès. En utilisant les notations introduites dans la section 5.1.2, nous pouvons modéliser notre problème sous la forme du programme mathématique suivant :

$$(\mathcal{LSR}) \left\{ \begin{array}{l} \min_{\mathbf{x}, \mathbf{r}, \mathbf{s}, \mathbf{m}} \sum_{i=1}^{\mathbf{m}} \sum_{j=1}^n \mathbf{x}_{ij} ((w_j A_1 + B_1)d(i, j) + w_j A_2 d(i, NRA)) \\ \quad + \sum_{i=1}^{\mathbf{m}} B_2 d(i, NRA) + \mathbf{m}C \\ \text{s.c.} \\ \quad d(i, j) = \sqrt{(\mathbf{r}_i - a_j)^2 + (\mathbf{s}_i - b_j)^2} \quad \forall i = 1 \dots m; \forall j = 1 \dots n \\ \quad d(i, NRA) = \sqrt{(\mathbf{r}_i - c)^2 + (\mathbf{s}_i - d)^2} \quad \forall i = 1 \dots m \\ \quad \sum_{i=1}^{\mathbf{m}} \mathbf{x}_{ij} = 1 \quad \forall j = 1 \dots n \\ \quad \mathbf{x}_{ij} \in \{0, 1\} ; \mathbf{r}, \mathbf{s} \in \mathbb{R}^{\mathbf{m}} ; \mathbf{m} \in \mathbb{N} \end{array} \right. \quad (5.3)$$

- n est le nombre de clients ;
- \mathbf{m} est le nombre d'équipements ;
- (a_j, b_j) sont les coordonnées du client j ;
- (c, d) sont les coordonnées du NRA ;

- \mathbf{x}_{ij} est une variable de décision qui vaut 1 lorsque le client j est raccordé au sous-répartiteur i , 0 sinon ;
- \mathbf{X} est la matrice des x_{ij} ;
- $(\mathbf{r}_i, \mathbf{s}_i)$ sont les coordonnées de l'équipement i ;
- Les contraintes (5.3) permettent de s'assurer que chaque client est raccordé à un et un seul équipement.

Dans l'objectif, on peut voir que le coût associé à l'affectation d'un client j à un SR i prend en compte à la fois la totalité du coup de liaison entre ce client et ce SR ($(w_j A_1 + B_1)d(i, j)$), mais aussi la contribution du client au coût de capacité de la liaison SR-NRA ($w_j A_2 d(i, NRA)$). Dans les autres termes de l'objectif, on trouve les coûts d'infrastructure des liaisons SR-NRA, et les coûts d'installation des SR.

Cette modélisation fait apparaître les deux sources principales de difficulté du problème : le fait de ne pas connaître le nombre d'équipement \mathbf{m} à placer d'une part, et la présence de la distance euclidienne d'autre part. Il est intéressant de noter que si le coût de la liaison SR-NRA était indépendant des clients rattachés au SR, le modèle deviendrait celui d'un problème de Weber multi-source avec nombre d'équipements inconnus (voir chapitre 4), où le NRA est juste un client particulier devant être raccordé à tous les équipements.

Notre problème est une généralisation du problème de Weber multi-source avec nombre d'équipements inconnu. Il est donc lui aussi NP-Difficile. Un autre point commun est que si les emplacements des SR (respectivement les affectations des clients au SR) sont connus, le problème devient facile :

- Si les vecteurs (\mathbf{r}, \mathbf{s}) des coordonnées des équipements sont connus, l'allocation des clients se fait simplement au SR de moindre coût :

$$\mathbf{x}_{ij} = 1 \Leftrightarrow i = \arg \min_{1 \leq k \leq m} \left((w_j A_1 + B_1) \sqrt{(r_k - a_j)^2 + (s_k - b_j)^2} + w_j A_2 \sqrt{(r_k - c)^2 + (s_k - d)^2} \right) \quad (5.4)$$

- Si la matrice X des variables d'affectation est fixée, alors chaque SR i est localisé en résolvant le problème de Weber simple suivant :

$$\min_{(\mathbf{r}_i, \mathbf{s}_i)} \sum_{j|x_{ij}=1} (w_j A_1 + B_1) \sqrt{(r_k - a_j)^2 + (s_k - b_j)^2} + \left(\sum_{j|x_{ij}=1} w_j A_2 + B_2 \right) \sqrt{(r_k - c)^2 + (s_k - d)^2} \quad (5.5)$$

Cette propriété nous permet notamment d'utiliser l'heuristique LOCALLOC de Cooper [10] (voir section 4.2) sur notre problème.

Pour éviter que la variable \mathbf{m} représentant le nombre d'équipements n'apparaissent dans les indices des sommes, il est possible de reformuler le modèle en prenant en compte le fait qu'il ne peut y avoir plus d'un SR par client. On considère alors n SR potentiels et on choisit lesquels seront utilisés :

$$(\mathcal{LSR}_{bis}) \left\{ \begin{array}{l} \min_{\mathbf{x}, \mathbf{y}, \mathbf{r}, \mathbf{s}} \quad \sum_{i=1}^n \sum_{j=1}^n \mathbf{x}_{ij} ((w_j A_1 + B_1)d(i, j) + w_j A_2 d(i, NRA)) \\ \quad \quad \quad + \sum_{i=1}^n \mathbf{y}_i (B_2 d(i, NRA) + C) \\ \text{s.c.} \\ d(i, j) = \sqrt{(\mathbf{r}_i - a_j)^2 + (\mathbf{s}_i - b_j)^2} \quad \forall i = 1 \dots n; \forall j = 1 \dots n \\ d(i, NRA) = \sqrt{(\mathbf{r}_i - c)^2 + (\mathbf{s}_i - d)^2} \quad \forall i = 1 \dots n \\ \sum_{i=1}^n \mathbf{x}_{ij} = 1 \quad \forall j = 1 \dots n \quad (5.6) \\ \mathbf{x}_{ij} \leq \mathbf{y}_i \quad \forall i = 1 \dots n; \forall j = 1 \dots n \quad (5.7) \\ \mathbf{x}_{ij} \in \{0, 1\} ; \mathbf{y}_i \in \{0, 1\} ; \mathbf{r}, \mathbf{s} \in \mathbb{R}^n \end{array} \right.$$

- \mathbf{y}_i est une variable de décision qui vaut 1 lorsque le sous-répartiteur i est installé, 0 sinon ;
- Les contraintes (5.8) ont le même but que les contraintes (5.3) du problème (\mathcal{LSR}) ;
- Les contraintes (5.9) évitent de rattacher un client à un SR qui n'est pas installé.

Cette modélisation est plus satisfaisante mais elle présente un grand nombre de symétries, rien ne différenciant les SR entre eux. De plus, la fonction objectif, comme celle du problème de Weber multi-source, n'est ni linéaire, ni quadratique, ni convexe ni concave et le modèle ne peut donc être passé tel quel à un solveur.

5.3 Comparaison entre géométrie stochastique et heuristiques

Le problème étant une généralisation du problème de Weber multi-source avec nombre d'équipements inconnu, nous avons utilisé le même type de méthodes pour le résoudre. Nous avons d'abord adapté l'heuristique multi-phase de [7]. Cette heuristique implique de résoudre le problème de localisation discrète associé à notre problème. Malheureusement, la taille de nos instances peut rendre cette partie difficile. Nous avons donc cherché à développer des heuristiques plus simples et rapides :

une méta-heuristique de recuit-simulé, et une heuristique basée sur LOCALLOC et sur la géométrie stochastique. Ces deux méthodes de résolution ont été initialement développées pour notre problème de localisation de sous-répartiteurs mais peuvent être utilisées pour le problème de Weber multi-source avec nombre d'équipements inconnus. Elles ont donc été présentées dans les sections 4.1.2, 4.2, et 4.3 du chapitre précédent. Pour obtenir des bornes inférieures afin de juger de la qualité des solutions retournées par les heuristiques, nous nous sommes tournés vers la génération de colonnes.

Dans cette section sont présentés l'ensemble de ces méthodes de résolution et leurs résultats numériques.

5.3.1 Géométrie stochastique

Une première méthode de résolution consiste à utiliser uniquement la modélisation sous forme de processus ponctuels. Grâce à la formule de [1] explicitée dans la section 5.2.1, nous calculons l'intensité optimale du processus ponctuel de SR à partir de celle des NRA et des clients. Une fois cette intensité obtenue, un nombre arbitraire de réalisations sont générées. Le raccordement des clients est alors fait au moindre coût, et le tirage de meilleur coût est conservé comme solution.

5.3.2 Heuristique multi-phase

Il s'agit ici d'adapter l'heuristique multi-phase développée dans [7] pour le problème de Weber multi-source avec nombre d'équipements inconnu. La première phase de l'heuristique consiste à résoudre le problème discret associé à notre problème. Pour cela, prenons le modèle (\mathcal{LSR}_{bis}) et fixons les vecteurs (\mathbf{r}, \mathbf{s}) aux coordonnées des clients (*i.e.* les emplacements potentiels des SR seront ceux des clients). Les quantités $d(i, j)$ et $d(i, NRA)$ deviennent alors des constantes connues. On obtient donc le modèle suivant :

$$(\mathcal{LSR}_{discret}) \left\{ \begin{array}{l} \min_{\mathbf{x}, \mathbf{y}} \quad \sum_{i=1}^n \sum_{j=1}^n \mathbf{x}_{ij} c_{ij} + \sum_{i=1}^n \mathbf{y}_i f_i \\ \text{s.c.} \\ \sum_{i=1}^n \mathbf{x}_{ij} = 1 \quad \forall j = 1 \dots n \quad (5.8) \\ \mathbf{x}_{ij} \leq \mathbf{y}_i \quad \forall i = 1 \dots n; \forall j = 1 \dots n \quad (5.9) \\ \mathbf{x}_{ij} \in \{0, 1\} ; \mathbf{y}_i \in \{0, 1\} \end{array} \right. \quad (5.8)$$

où :

$$\begin{aligned}c_{ij} &= ((w_j A_1 + B_1)d(i, j) + w_j A_2 d(i, NRA)) \\ f_i &= (B_2 d(i, NRA) + C)\end{aligned}$$

Il s'agit, comme pour le problème de Weber multi-source avec nombre d'équipements inconnu, d'un problème de localisation simple. La seule différence est qu'ici, le coût d'installation d'un équipement en i n'est pas identique pour tout i .

Ce modèle sera résolu de façon approchée ou de façon exacte par des méthodes classiques de résolution du problème de localisation simple afin d'obtenir une première solution de localisation des SR et d'affectation des clients. Dans la deuxième phase, l'heuristique LOCALLOC est appliquée à cette solution. Rappelons que cette heuristique consiste à alterner entre phases de localisation, où chaque SR est localisé de façon optimale par rapport à ses clients, et phase d'allocation, où les clients sont réalloués au SR de moindre coût. Nous n'avons pas choisi d'implémenter la troisième phase optionnelle de l'heuristique multi-phase qui consisterait à faire varier le nombre d'équipements.

Dans notre implémentation de l'heuristique multi-phase, le problème de localisation simple est résolu par un branch & bound basé sur la méthode de [22]. Une limite de temps pour la résolution du problème de localisation simple est fixée à 3 heures.

5.3.3 Heuristiques hybrides : géométrie stochastique et LOCALLOC

La comparaison des méthodes de recherche opérationnelle et de géométrie stochastique pour notre problème nous a amené à associer les deux méthodes, avec pour objectif de trouver un meilleur compromis entre qualité des solutions et temps de calcul. Ainsi, d'un côté l'utilisation de réalisations de processus ponctuel en remplacement de la résolution du problème de localisation simple permet de gagner en temps de calcul, et de l'autre côté l'utilisation de LOCALLOC sur ces réalisations permettent d'améliorer la répartition de leurs SR, et donc leur qualité en tant que solution.

Il y a deux possibilités d'utilisation des tirages aléatoires, qui aboutissent à deux méthodes hybrides différentes. Les deux méthodes commencent donc par générer un ensemble de solutions potentielles à l'aide de l'approche de géométrie stochastique. La différence apparaît une fois ces tirages effectués.

- La première méthode consiste à déterminer le tirage de meilleur coût pour une instance donnée, puis à améliorer ce tirage en utilisant LOCALLOC ;
- La deuxième méthode applique LOCALLOC à l'ensemble des tirages de SR effectués, et ensuite détermine le meilleur.

Évidemment, la première méthode est bien plus rapide que la seconde, dont le temps de calcul dépend directement du nombre de solutions générées par la géométrie stochastique. La deuxième méthode, quant à elle, a plus de chances de trouver de meilleures solutions.

5.3.4 Recuit simulé

N'ayant pas encore de borne sur la qualité des solutions trouvées, nous avons voulu comparer l'heuristique multi-phase à plusieurs autres méthodes. Aux heuristiques hybrides nous avons donc ajouté une méthode de recuit simulé pour résoudre notre problème.

Une solution à notre problème sera décrite par l'ensemble des groupes de clients qui sont rattachés au même sous-répartiteur. On effectue donc une partition de l'ensemble des clients, le nombre de classes représentant alors le nombre de sous-répartiteurs. Pour les détails, nous renvoyons le lecteur à la section 4.3, le principe et les mouvements pratiqués sur la solution étant exactement les mêmes que pour le problème de Weber multi-source avec nombre d'équipements inconnu.

5.3.5 Obtention de bornes inférieures par génération de colonnes

Aucune des heuristiques présentées ci-dessus ne permet d'obtenir une borne inférieure de la valeur optimale, nous ignorons la qualité des solutions obtenues. Pour remédier à cela, nous nous sommes tournés vers la génération de colonnes qui à défaut de fournir des solutions exactes à l'aide d'un branch & price, pouvait tout au moins nous fournir des bornes inférieures en résolvant la relaxation continue du problème. La taille des instances et les problèmes de dégénérescence et de stabilisation rencontrés nous ont poussé à chercher à améliorer la stabilité et à accélérer le plus possible la génération de colonnes. C'est dans ce but qu'a été développée la méthode de génération de colonnes dite « centrale » présentée au chapitre 2.

5.3.5.1 Modélisation

Dans l'heuristique multi-phase, nous commençons par discrétiser le problème en prenant les sites des clients comme emplacements potentiels pour les SR. Cette discrétisation permet de reléguer la distance entre un client et son SR à une simple donnée connue, puisqu'elle dépend non plus des coordonnées variables de ce SR, mais uniquement de l'emplacement potentiel choisi. Les distances peuvent donc toutes être calculées a priori.

Malheureusement, dans le problème initial continu nous n'avons pas de tels emplacements potentiels... à première vue. Mais de tels emplacements existent pourtant. En

effet, nous avons vu que la localisation d'un SR se déduisait simplement une fois ses clients connus. Nous savons donc qu'à l'optimum les sous-répartiteurs occupent les emplacement de points qui sont solution d'un problème de Weber simple prenant en compte leurs clients et le NRA. Or ces points sont en nombre fini. À chaque groupe de clients va correspondre un seul de ces points. Nous avons donc en fait autant d'emplacements potentiels que de regroupements possibles de clients, soit $2^n - 1$ points. À chacun de ces points va correspondre un coût correspondant à l'installation du SR et aux liaisons entre le NRA, le SR et les clients attribués à ce point. Le but du problème est alors de choisir parmi ces emplacements potentiels ceux qui donnent le coût le plus faible tout en couvrant tous les clients. Cela se traduit par le modèle de couverture d'ensemble suivant :

$$(\mathcal{LSR}_{setcovering}) \left\{ \begin{array}{l} \min_{\mathbf{z}} \sum_{i=1}^{2^n-1} c^i \mathbf{z}_i \\ \text{s.c.} \\ \sum_{i=1}^{2^n-1} x_j^i \mathbf{z}_i \geq 1 \quad \forall j = 1 \dots n \\ \mathbf{z}_i \in \{0, 1\} \end{array} \right. \quad (5.10)$$

- \mathbf{z}_i est une variable de décision qui vaut 1 lorsque le groupe de clients d'indice i est utilisé, 0 sinon ;
- c^i est le coût total associé à l'installation d'un SR pour le groupe de clients i . Il comprend le coût d'installation lui même plus le coût optimal du problème de Weber simple optimisant le placement d'un SR pour ce groupe de clients. Ce coût c^i a donc la valeur suivante :

$$C + \min_{(\mathbf{r}_i, \mathbf{s}_i)} \left[\sum_{j=1}^n x_j^i (w_j A_1 + B_1) \sqrt{(\mathbf{r}_i - a_j)^2 + (\mathbf{s}_i - b_j)^2} \right. \\ \left. + \left(\sum_{j=1}^n x_j^i w_j A_2 + B_2 \right) \sqrt{(\mathbf{r}_i - c)^2 + (\mathbf{s}_i - d)^2} \right] \quad (5.11)$$

- x_j^i est la valeur de la fonction indicatrice du groupe de clients i pour le client j (*i.e* x_j^i vaut 1 si j est dans l'ensemble i , 0 sinon) ;
- Les contraintes (5.10) permettent de s'assurer que tous les clients sont bien dans au moins une partie. Cela revient à ce que chaque client soit bien relié à un SR.

Ce type de formulation est la même que celle utilisée pour le problème de Weber multi-source où le nombre d'équipements à placer est connu (voir à ce sujet [23] [39])

Propriété 5.1. *Il existe une solution optimale (\mathbf{x}^*, r^*, s^*) du sous-problème (SP_{LSR}) , pour laquelle on a :*

$$\begin{aligned} \forall j = 1 \dots n, x_j^* = 1 \Leftrightarrow & (w_j A_1 + B_1) \sqrt{(r^* - a_j)^2 + (s^* - b_j)^2} \\ & + w_j A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_j \leq 0 \end{aligned} \quad (5.12)$$

Démonstration.

- Dans le sens \Rightarrow :

Supposons que la propriété ci-dessus n'est pas vérifiée pour $j = n$ (l'indice peut être choisi arbitrairement sans perte de généralité), et que nous avons :

$$\begin{cases} x_n^* = 1 \\ (w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n > 0 \end{cases}$$

En gardant les valeurs des x_j^* , en fixant x_n^* à 0, et en relocalisant le SR au mieux, on obtient une solution $(\hat{\mathbf{x}}, \hat{r}, \hat{s})$ qui vérifie :

$$\hat{x}_j = \begin{cases} x_j^* & \forall j \neq n \\ 0 & \text{pour } j = n \end{cases}$$

Et :

$$\begin{aligned} \bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) = \min_{(\mathbf{r}, \mathbf{s})} & \left\{ \sum_{j=1}^{n-1} \left[x_j^* (w_j A_1 + B_1) \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} + w_j A_2 \sqrt{(\mathbf{r} - c)^2 + (\mathbf{s} - d)^2} - u_j \right] \right. \\ & \left. + B_2 \sqrt{(\mathbf{r} - c)^2 + (\mathbf{s} - d)^2} \right\} \end{aligned}$$

En particulier :

$$\begin{aligned} \bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) & \leq \sum_{j=1}^{n-1} x_j^* \left[(w_j A_1 + B_1) \sqrt{(r^* - a_j)^2 + (s^* - b_j)^2} + w_j A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_j \right] \\ & \quad + B_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} \\ \Leftrightarrow \bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) & \leq \bar{c}(\mathbf{x}^*, r^*, s^*) - x_n^* \left((w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} \right. \\ & \quad \left. + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n \right) \end{aligned}$$

Par hypothèse :

$$x_n^* \left((w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n \right) > 0$$

Donc :

$$\bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) < \bar{c}(\mathbf{x}^*, r^*, s^*)$$

Ce qui est en contradiction avec le fait que (\mathbf{x}^*, r^*, s^*) soit une solution optimale.

- Dans le sens \Leftarrow :

Supposons maintenant que nous avons :

$$\begin{cases} x_n^* = 0 \\ (w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n \leq 0 \end{cases}$$

En gardant les valeurs des x_j^* , en fixant x_n^* à 1, et en relocalisant le SR au mieux, on obtient une solution $(\hat{\mathbf{x}}, \hat{r}, \hat{s})$ qui vérifie :

$$\hat{x}_j = \begin{cases} x_j^* & \forall j \neq n \\ 1 & \text{pour } j = n \end{cases}$$

Et :

$$\begin{aligned} \bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) = \min_{(\mathbf{r}, \mathbf{s})} & \left\{ \sum_{j=1}^{n-1} x_j^* \left((w_j A_1 + B_1) \sqrt{(\mathbf{r} - a_j)^2 + (\mathbf{s} - b_j)^2} + w_j A_2 \sqrt{(\mathbf{r} - c)^2 + (\mathbf{s} - d)^2} - u_j \right) \right. \\ & + \left((w_n A_1 + B_1) \sqrt{(\mathbf{r} - a_n)^2 + (\mathbf{s} - b_n)^2} + w_n A_2 \sqrt{(\mathbf{r} - c)^2 + (\mathbf{s} - d)^2} - u_n \right) \\ & \left. + B_2 \sqrt{(\mathbf{r} - c)^2 + (\mathbf{s} - d)^2} \right\} \end{aligned}$$

En particulier :

$$\begin{aligned} \bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) & \leq \sum_{j=1}^{n-1} x_j^* \left((w_j A_1 + B_1) \sqrt{(r^* - a_j)^2 + (s^* - b_j)^2} + w_j A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_j \right) \\ & + \left((w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n \right) \\ & + B_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} \\ \Leftrightarrow \bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) & \leq \left((w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n \right) \\ & + \bar{c}(\mathbf{x}^*, r^*, s^*) \end{aligned}$$

Par hypothèse :

$$(w_n A_1 + B_1) \sqrt{(r^* - a_n)^2 + (s^* - b_n)^2} + w_n A_2 \sqrt{(r^* - c)^2 + (s^* - d)^2} - u_n \leq 0$$

Donc :

$$\bar{c}(\hat{\mathbf{x}}, \hat{r}, \hat{s}) \leq \bar{c}(\mathbf{x}^*, r^*, s^*)$$

En cas d'égalité, $(\hat{\mathbf{x}}, \hat{r}, \hat{s})$ est une solution optimale. Sinon, il y a contradiction avec le fait que (\mathbf{x}^*, r^*, s^*) soit une solution optimale. \square

Avec la propriété 5.1, tout se passe comme si un client j devenait inintéressant dès que la somme de son coût de raccordement au SR et de sa contribution au coût de liaison SR-NRA dépassait un seuil fixé par la valeur duales u_j . Dans le problème de Weber avec distances maximales, un client j n'est intéressant que si l'équipement est situé à l'intérieur du cercle centré sur ce client et de rayon $\frac{u_j}{w_j}$. En revanche, dans la propriété 5.1 nous avons affaire non plus à des cercles mais à des sortes d'ellipses dont les foyers sont pondérés différemment. Ces courbes sont représentées par l'équation (5.13) et se nomment ovales de Descartes (voir figure 5.4). Ces courbes vont de l'ellipse au cercle centré sur l'un des deux foyers, selon le rapport des poids des deux foyers.

$$(w_j A_1 + B_1)d(SR, j) + w_j A_2 d(SR, NRA) = u_j \quad (5.13)$$

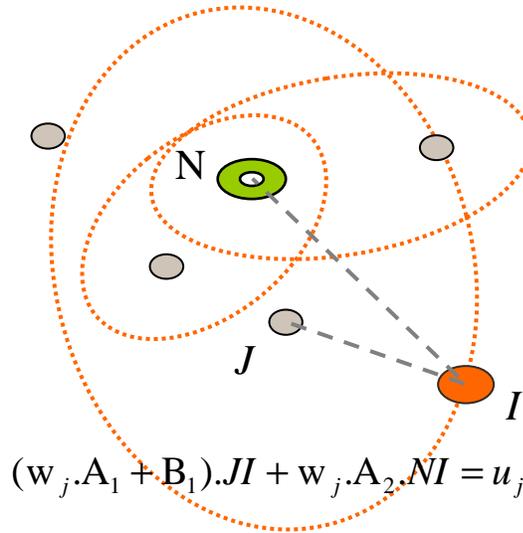
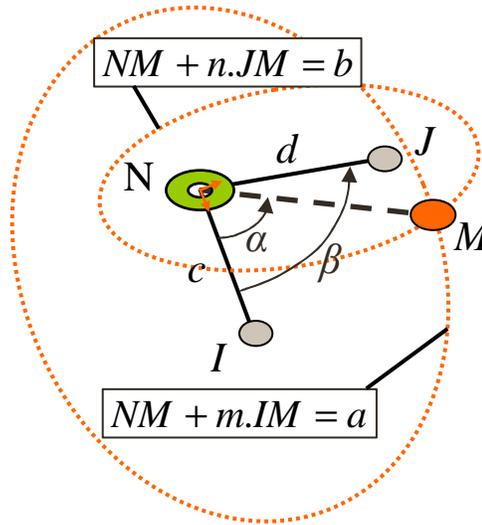


FIGURE 5.4 – Représentation par des ellipses des ovales de Descartes induits par la propriété 5.1

L'ovale de Descartes associé au client j a pour foyer le NRA et le client j . La propriété 5.1 signifie donc qu'il est possible d'associer une allocation optimale des clients à chaque région du plan découpé par les ovales de Descartes associés aux clients. L'idée pour résoudre notre sous-problème serait alors de procéder comme pour le

FIGURE 5.5 – Intersection de deux ovales de Descartes partageant le foyer N

problème de Weber avec distances maximales : énumérer ces régions à partir des points d'intersections des ovales et évaluer les groupes de clients ainsi obtenus.

Malheureusement, les points d'intersection de deux ovales de Descartes sont plus difficiles à caractériser que ceux de deux cercles. Mais cela reste faisable. Il est donc possible d'énumérer les solutions potentiellement optimales définies par ces ovales, à partir de leurs points d'intersection en s'inspirant de l'algorithme 4.2 . On note respectivement N , I , et J , les points représentant le NRA et deux clients i et j . Soit M un point appartenant aux deux ovales (voir figure 5.5). M vérifie les propriétés suivantes :

$$\begin{cases} w_i A_2 NM + (w_i A_1 + B_1) IM = u_i \\ w_j A_2 NM + (w_j A_1 + B_1) JM = u_j \end{cases}$$

$$\Leftrightarrow \begin{cases} NM + \frac{w_i A_1 + B_1}{w_i A_2} IM = \frac{u_i}{w_i A_2} \\ NM + \frac{w_j A_1 + B_1}{w_j A_2} JM = \frac{u_j}{w_j A_2} \end{cases}$$

Introduisons les notations suivantes :

- $m = \frac{w_i A_1 + B_1}{w_i A_2}$; $n = \frac{w_j A_1 + B_1}{w_j A_2}$;
- $a = \frac{u_i}{w_i A_2}$; $b = \frac{u_j}{w_j A_2}$;
- $c = NI$; $d = NJ$;
- $\alpha = (\widehat{N, I, M})$; $\beta = (\widehat{N, I, J})$;
- $r = NM$.

En utilisant le théorème d'Al-Kashi pour exprimer IM et JM , le système précédent devient :

$$\begin{aligned}
& \begin{cases} r + m\sqrt{r^2 + c^2 - 2rc \cos \alpha} = a \\ r + n\sqrt{r^2 + d^2 - 2rd \cos(\beta - \alpha)} = b \end{cases} \\
\Leftrightarrow & \begin{cases} (r - a)^2 = m^2 (r^2 + c^2 - 2cr \cos \alpha) \\ (r - b)^2 = n^2 (r^2 + d^2 - 2rd \cos(\beta - \alpha)) \end{cases} \\
\Leftrightarrow & \begin{cases} (r - a)^2 = m^2 (r^2 + c^2 - 2cr \cos \alpha) \\ (r - b)^2 = n^2 (r^2 + d^2 - 2rd (\cos \alpha \cos \beta + \sin \alpha \sin \beta)) \end{cases} \\
\Leftrightarrow & \begin{cases} (r - a)^2 = m^2 (r^2 + c^2 - 2cr \cos \alpha) \\ (r - b)^2 - n^2 (r^2 + d^2 - 2d \cos \beta r \cos \alpha) = -2dn^2 \sin \beta r \sin \alpha \end{cases} \\
\Leftrightarrow & \begin{cases} (r - a)^2 = m^2 (r^2 + c^2 - 2cr \cos \alpha) \\ \left((r - b)^2 - n^2 (r^2 + d^2 - 2d \cos \beta r \cos \alpha) \right)^2 = 4d^2 n^4 \sin^2 \beta r^2 \sin^2 \alpha \end{cases} \\
\Leftrightarrow & \begin{cases} (r - a)^2 = m^2 (r^2 + c^2 - 2cr \cos \alpha) \\ \left((r - b)^2 - n^2 (r^2 + d^2 - 2d \cos \beta r \cos \alpha) \right)^2 = 4d^2 n^4 \sin^2 \beta (r^2 - (r \cos \alpha)^2) \end{cases}
\end{aligned}$$

En notant x et y les coordonnées du point M dans le repère orthonormé centré en N , et dont l'axe des abscisses a la direction de \vec{NI} , nous pouvons remplacer $r \cos \alpha$ par x . Nous obtenons alors :

$$\begin{aligned}
& \begin{cases} (r - a)^2 = m^2 (r^2 + c^2 - 2cx) \\ \left((r - b)^2 - n^2 (r^2 + d^2 - 2d \cos \beta x) \right)^2 = 4d^2 n^4 \sin^2 \beta (r^2 - (x)^2) \end{cases} \\
\Leftrightarrow & \begin{cases} x = \frac{m^2(r^2 + c^2) - (r - a)^2}{2cm^2} & (5.14) \\ \left((r - b)^2 - n^2 (r^2 + d^2 - 2d \cos \beta x) \right)^2 = 4d^2 n^4 \sin^2 \beta (r^2 - x^2) & (5.15) \end{cases}
\end{aligned}$$

En remplaçant x dans l'équation (5.15) par son expression dans (5.14), nous obtenons une équation de degré 4 d'inconnue r . Cette équation a au maximum 4 solutions réelles positives (r est une distance), que l'on peut obtenir par exemple avec la méthode de Ferrari. Pour chacune de ces solutions nous pouvons calculer une abscisse x à laquelle peuvent correspondre deux ordonnées, y et $-y$ telles que : $y^2 = r^2 - x^2$. Pour chacun des deux points ainsi obtenus il faudra donc vérifier qu'il appartient bien au deux ovales de Descartes. Nous avons donc au maximum 8 couples (x, y) différents solutions du système ci-dessus.

Pour deux ovales de Descartes, nous obtenons donc au maximum 8 points d'intersections, au lieu de 2 pour deux cercles. Un raisonnement similaire à celui tenu

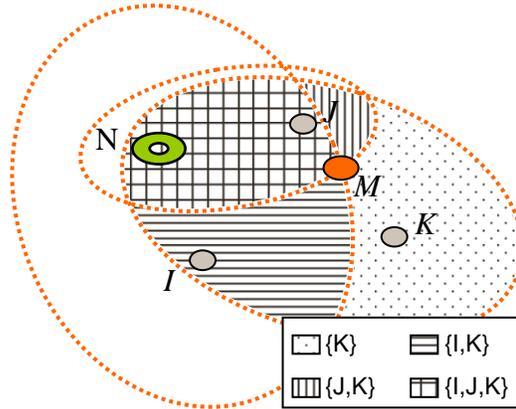


FIGURE 5.6 – Les 4 régions du plan auxquelles appartient le point d’intersection M et les groupes de clients correspondants.

dans [15] nous permet de déduire que le nombre de régions formées par le découpage du plan par les ovales de Descartes des clients est borné par $8n(n-1)$. En effet, si tous les ovales s’intersectent, le nombre de points d’intersection est alors : $\sum_{i=1}^{n-1} 8i = 4n(n-1)$. Chacun de ces points d’intersection appartient au maximum à 4 régions du plan (comme illustré dans la figure 5.6), et chacune des régions est définie par au moins 2 points d’intersection, d’où la borne de $8n(n-1)$.

Nous savons maintenant que nous avons juste à énumérer un nombre polynomial de groupes de clients potentiellement optimaux. Chacun de ces groupes sera évalué en résolvant un problème de Weber simple, comme dans l’algorithme de [15]. La complexité d’un tel algorithme pour notre sous-problème sera donc polynomiale en le nombre de clients et le temps de résolution d’un problème de Weber simple (méthode de descente du gradient).

Il est intéressant de remarquer que nous disposons d’une information supplémentaire sur les ovales de Descartes par rapport aux cercles, à savoir une condition d’existence :

$$\min \{w_j A_1 + B_1; w_j A_2\} d(j, NRA) \leq u_j \quad (5.16)$$

Si un client j ne vérifie pas (5.16), alors on peut éliminer toutes les solutions où il est présent. Ces tests peuvent s’effectuer rapidement et permettent de réduire l’ensemble des solutions explorées. Nous avons toutefois constaté que cette réduction diminuait au fur et à mesure du déroulement de l’algorithme de génération de colonnes.

Étant capable de résoudre le sous-problème à l’exact, nous avons pu implémenter les deux méthodes de générations de colonnes présentées aux chapitres 1 et 2.

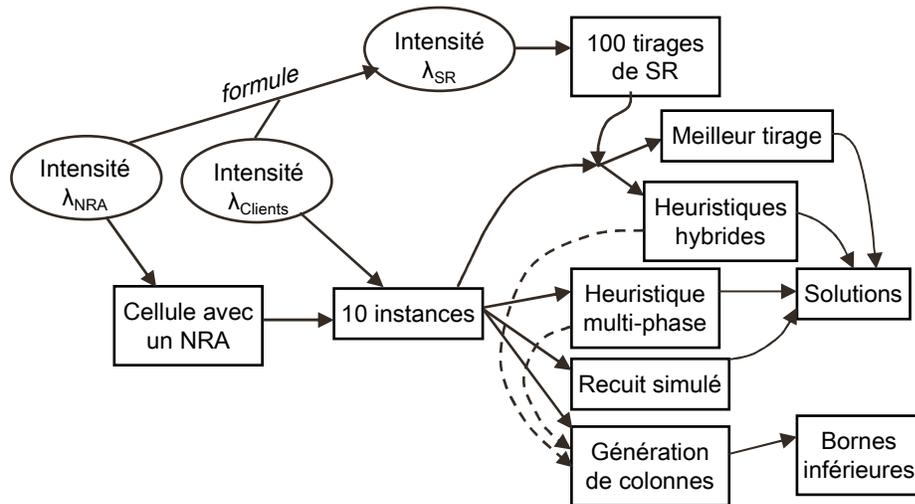


FIGURE 5.7 – Méthodologie des tests sur les instances artificielles

5.3.6 Comparaison sur des instances générées aléatoirement

Les données d'instances réelles étant difficiles à obtenir, et afin de faciliter la comparaison avec les méthodes de géométrie stochastique, nous avons généré des ensembles d'instances artificielles à partir de processus ponctuels.

Les tests des heuristiques et du recuit simulé ont été réalisés sur un serveur IBM XSERIES_455 disposant de quatre processeurs cadencés à 3Ghz (un seul est utilisé) et de 10Go de mémoire vive. Les tests des générations de colonnes ont quant eux été réalisés sur un serveur linux équipé d'un processeur xeon 3,6Ghz et de 4Go de mémoire vive.

5.3.6.1 Méthodologie des tests

La méthodologie des tests est représentée sur la figure 5.7. Un processus ponctuel de Poisson modélise les NRA et un autre les clients. Celui modélisant les NRA est utilisé pour générer un pavage de Voronoï (voir la définition 4.1 page 74) dont est extrait une cellule type. Cette cellule type est utilisée comme zone locale. Le processus ponctuel de Poisson modélisant les clients est alors utilisé pour générer plusieurs ensembles de clients à l'intérieur de la cellule type. On obtient ainsi plusieurs instances différentes correspondant à un couple de densités moyennes de NRA et de clients.

Les différentes méthodes de résolution ou d'obtention de borne inférieure sont ensuite utilisées sur ces instances. Pour les méthodes utilisant la géométrie stochastique (approche de géométrie stochastique pure et heuristiques hybrides), l'intensité opti-

male du processus modélisant les SR est calculée et un certain nombre d'ensemble de SR sont générés pour être utilisés dans ces approches.

La génération de colonnes est initialisées à l'aide des colonnes des solutions issues de l'heuristique multi-phase et des heuristiques hybrides.

5.3.6.2 Résultats numériques

Les résultats présentés dans cette section sont organisés en deux parties. D'abord nous présentons les résultats obtenus par l'heuristique multi-phase sur des instances de taille moyennes, comparés à la borne inférieure obtenue par génération de colonnes. Les instances utilisées sont celles ayant servi à comparer les deux générations de colonnes dans la section 2.5. Il s'agit de deux séries de dix instances avec une moyenne d'environ 500 clients pour la première série, et de 725 clients pour la seconde. Les résultats sont présentés dans le tableau 5.1.

Il est indiqué pour chaque instance : le nombre de clients, la valeur de la borne inférieure obtenue par génération de colonnes, la valeur de la solution de l'heuristique multi-phase, le temps CPU utilisé par la génération de colonne centrale, l'écart relatif entre la solution heuristique et la borne inférieure, le temps CPU utilisé par l'heuristique multi-phase, et le nombre de SR utilisés dans la solution de la relaxation continue et dans celle de l'heuristique multi-phase.

Ces résultats montre l'efficacité de l'heuristique multi-phase sur les deux ensembles d'instances. L'écart relatif moyen avec la borne inférieure reste inférieur à 0,1%, et le temps de calcul ne dépasse pas la minute. Les résultats sur les instances pour lesquelles la génération de colonnes obtient la solution optimale permettent de constater que le nombre d'équipements présents dans les solutions de l'heuristique est très proche du nombre optimal, avec une différence maximum de 1.

Le temps de calcul de la génération de colonnes est déjà très long sur ces instances de taille moyenne. Par conséquent, ayant montré sur ces instances la qualité des solutions de l'heuristique multi-phase, dans la suite des résultats les solutions des différentes méthodes seront comparées uniquement à celle de l'heuristique multi-phase, sans borne inférieure.

Les tableaux 5.2, 5.3, 5.4, et 5.5 présentent les résultats obtenus par les différentes méthodes d'optimisation présentées dans ce chapitre, sur des instances allant de 1189 à 3092 clients. Ces instances sont regroupées en quatre ensemble de dix instances de taille similaire. Sont présentés pour chaque méthode et chaque instance la valeur de la solution obtenue, le temps de calcul en secondes, et l'écart relatif avec la valeur de la solution de l'heuristique multi-phase.

Pour l'heuristique multi-phase, nous avons utilisé pour la première phase la méthode primale-duale de [22]. Le temps alloué au branch & bound pour résoudre le problème

Instance	0	1	2	3	4	5	6	7	8	9	Moyenne
Taille	515	508	530	492	561	468	536	504	562	485	516.1
Borne inférieure	391859	392855	400463	376750	414670	363457	393657	383986	413315	370954	
Solution heuristique	392341	393124	400786	377084	415052	363657	393704	384002	413592	371095	
Temps CPU Gén. Col. (s)	555	627	495	691	853	575	568	433	751	519	606.7
Écart de l'heuristique	0.12%	0.07%	0.08%	0.09%	0.09%	0.06%	0.01%	0.004%	0.07%	0.04%	0.06%
Temps CPU heuristique (s)	6	1	3	13	4	6	7	2	7	4	5.3
Nbre de SR Gén. Col.	28	26.5 ¹	30	29 ¹	28.5 ¹	27	29	26	30	27 ¹	
Nbre de SR heuristique	28	27	31	29	28	27	29	26	30	27	
Instance	10	11	12	13	14	15	16	17	18	19	Moyenne
Taille	778	724	744	671	735	747	739	717	787	736	737.8
Borne inférieure	614671	577720	590511	553245	589612	595074	600231	574386	620545	594641	
Solution heuristique	614973	577962	591272	553497	590141	595242	600881	574816	621471	595261	
Temps CPU Gén. Col. (s)	1162	702	1059	945	1138	1430	1084	1068	1336	1071	1099.5
Écart de l'heuristique	0.05%	0.04%	0.13%	0.05%	0.09%	0.03%	0.11%	0.07%	0.15%	0.10%	0.08%
Temps CPU heuristique (s)	40	5	33	8	24	5	33	28	28	27	23.1
Nbre de SR Gén. Col.	41.67 ¹	39	41	38	38.71 ¹	39.5 ¹	40.5 ¹	38.5 ¹	39.67 ¹	41	
Nbre de SR heuristique	41	40	40	38	39	40	38	38	41	42	

TABLE 5.1 – Résultats de l'heuristique multi-phase sur des instances de taille moyenne

^a. La solution de la relaxation continue n'est pas entière.

	Instances Taille	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	Moyenne 1254
Heuristique multi-phase	Coût Tps CPU	3412792 318	3557940 483	3363414 347	3479381 498	3402685 299	3299309 306	3455140 332	3367153 145	3358448 138	3544953 634	350
Recuit simulé	Coût Écart Tps CPU	3557871 4.25% 501.76	3703951 4.10% 545.84	3473866 3.28% 490.12	3610788 3.78% 541.24	3537312 3.96% 500.16	3415839 3.53% 456.04	3587745 3.84% 541.56	3503759 4.06% 498.5	3494649 4.06% 479.98	3672859 3.61% 545.66	3.85% 510.09
Recuit +	Coût Écart Tps CPU	3579403 4.88% 87.96	3748952 5.37% 96.04	3523910 4.77% 84.44	3651619 4.95% 86.18	3596315 5.69% 93.66	3486368 5.67% 75.22	3625337 4.93% 97.2	3592553 6.69% 86.78	3512640 4.59% 77.18	3747265 5.71% 92.18	5.33% 87.68
Recuit +	Coût Écart Tps CPU	3517646 3.07% 78	3656162 2.76% 77.88	3464021 2.99% 76.74	3577430 2.82% 78.3	3508383 3.11% 78.48	3404413 3.19% 67.68	3566297 3.22% 80.4	3494183 3.77% 71.5	3454103 2.85% 70.78	3659698 3.24% 79.44	3.10% 75.92
Méthode hybride 1	Coût Écart Tps CPU	3544223 3.85% 81	3673419 3.25% 81	3502170 4.13% 81	3609032 3.73% 81	3512893 3.24% 81	3450497 4.58% 81	3559649 3.02% 81	3493708 3.76% 81	3503285 4.31% 81	3665914 3.41% 81	3.73% 81.00
Méthode hybride 2	Coût Écart Tps CPU	3480960 2.00% 646	3610470 1.48% 722	3422930 1.77% 657	3551900 2.08% 671	3471640 2.03% 666	3363740 1.95% 628	3535920 2.34% 705	3429360 1.85% 662	3432780 2.21% 633	3599390 1.54% 710	1.92% 670.00

TABLE 5.2 – Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 1250 clients

	Instances		PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	Moyenne
	Taille		1677	1671	1653	1684	1699	1686	1705	1713	1700	1621	1681
Heuristique multi-phase	Coût	4452164	4468108	4401096	4426545	4573129	4509870	4536400	4536765	4511475	4329543		1124.25
	Tps CPU	417	2426	184	11965.1	12214.1	977	1859	445	1245	1441		
Recuit simule	Coût	4693594	4722101	4632188	4657150	4785016	4732659	4757305	4767410	4744409	4534822		5.10%
	Écart	5.42%	5.68%	5.25%	5.21%	4.63%	4.94%	4.87%	5.08%	5.16%	4.74%		5.10%
	Tps CPU	889.14	892.8	865.06	907.82	931.06	912.28	935.04	932.6	913.46	832.56		901.18
Recuit + Localloc(2)	Coût	4724577	4714691	4704243	4731544	4813271	4788387	4807947	4807142	4764686	4537953		5.92%
	Écart	6.12%	5.52%	6.89%	6.89%	5.25%	6.18%	5.99%	5.96%	5.61%	4.81%		5.92%
	Tps CPU	150.92	157.64	153.86	156.86	155.78	169.36	156.56	171.3	165.1	156.42		159.38
Recuit + Localloc(2)	Coût	4621545	4622573	4590173	4600824	4722416	4687349	4707002	4696684	4677864	4438150		3.62%
	Écart	3.80%	3.46%	4.30%	3.94%	3.26%	3.94%	3.76%	3.52%	3.69%	2.51%		3.62%
	Tps CPU	130.42	138.62	132	134.34	141.9	143.74	135.34	142.42	148.42	136.42		138.36
Méthode hybride 1	Coût	4609051	4726567	4620750	4555421	4763406	4742073	4679897	4798429	4676317	4485695		4.27%
	Écart	3.52%	5.78%	4.99%	2.91%	4.16%	5.15%	3.16%	5.77%	3.65%	3.61%		4.27%
	Tps CPU	125	125	125	125	125	125	125	125	125	125		125.00
Méthode hybride 2	Coût	4507122	4550036	4505971	4529604	4641282	4591548	4597797	4631910	4597266	4419682		1.85%
	Écart	1.23%	1.83%	2.38%	2.33%	1.49%	1.81%	1.35%	2.10%	1.90%	2.08%		1.85%
	Tps CPU	992	990	953	1032	1038	993	1036	1024	981	938		997.70

TABLE 5.3 – Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 1680 clients

a. Le branch & bound a atteint sa limite de 10800 secondes.

Instances	PC20		PC21		PC22		PC23		PC24		PC25		PC26		PC27		PC28		PC29		Moyenne	
	Taille	2227	2179	2153	2261	2314	2346	2297	2231	2231	2237	2301	2255									
Heuristique multi-phase	Coût	5386089	5232242	5229038	5398014	5500759	5588782	5475327	5310393	5381735	5467253	5292.667										
	Tps CPU	17282 ¹	14587 ¹	5748	16660 ¹	4375	15869 ¹	5755	13248 ¹	14486 ¹	14203 ¹	5292.667										
Recuit simulé	Coût	5678775	5545615	5536783	5710904	5831682	5937689	5812420	5627418	5692401	5799258	5.93%										
	Écart	5.43%	5.99%	5.89%	5.80%	6.02%	6.24%	6.16%	5.97%	5.77%	6.07%	5.93%										
	Tps CPU	1194.28	1187.34	1179.2	1194.9	1205.02	1202.58	1205.9	1201.12	1202.56	1205.46	1197.84										
Recuit + Localloc(2)	Coût	5918661	5763405	5792406	5927156	6080530	6152225	6039667	5848916	5920195	6057031	10.25%										
	Écart	9.89%	10.15%	10.77%	9.80%	10.54%	10.08%	10.31%	10.14%	10.01%	10.79%	10.25%										
	Tps CPU	215.92	220.34	209.7	233.3	239.58	250.56	255.12	223.54	238.46	230.92	231.74										
Recuit + Localloc(inf)	Coût	5707925	5544292	5508134	5665703	5831983	5898384	5778383	5591611	5654796	5793865	5.57%										
	Écart	5.98%	5.96%	5.34%	4.96%	6.02%	5.54%	5.53%	5.30%	5.07%	5.97%	5.57%										
	Tps CPU	162.48	177.04	170.82	185.98	176.24	173.2	177.14	169.54	167.1	188.16	174.77										
Méthode hybride 1	Coût	5552603	5394531	5416979	5562739	5696358	5765668	5602084	5556213	5530385	5680571	3.32%										
	Écart	3.09%	3.10%	3.59%	3.05%	3.56%	3.17%	2.32%	4.63%	2.76%	3.90%	3.32%										
	Tps CPU	143	143	143	143	143	143	143	143	143	143	143.00										
Méthode hybride 2	Coût	5457761	5336490	5297846	5486496	5603705	5694172	5563282	5422560	5504891	5542794	1.74%										
	Écart	1.33%	1.99%	1.32%	1.64%	1.87%	1.89%	1.61%	2.11%	2.29%	1.38%	1.74%										
	Tps CPU	1820	1801	1720	1845	1902	1978	1919	1827	1813	1883	1850.80										

TABLE 5.4 – Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 2250 clients

a. Le branch & bound a atteint sa limite de 10800 secondes.

	Instances	PC30	PC31	PC32	PC33	PC34	PC35	PC36	PC37	PC38	PC39	Moyenne
	Taille	2940	2894	3092	2960	3023	2956	2960	2928	2861	2925	
Heuristique	Coût	6156754	6040159	6360328	6102689	6228584	6126774	6110998	6044634	5944400	6092508	14975
multi-phase	Tps CPU	27197 ¹	26626 ¹	28597 ¹	14975	25910 ¹	27169 ¹	24615 ¹	20594 ¹	24325 ¹	20034 ¹	
Recuit	Coût	6661103	6519534	6915034	6618863	6788502	6630316	6612506	6547517	6446578	6617507	8.36%
simule	Écart	8.19%	7.94%	8.72%	8.46%	8.51%	8.22%	8.21%	8.32%	8.45%	8.62%	
	Tps CPU	1209.28	1209.62	1209.7	1209.46	1210.22	1209.8	1209.64	1209.82	1208.74	1209.08	1209.54
Recuit	Coût	6808241	6650774	7027599	6753626	6961417	6803551	6795877	6710261	6575894	6784574	10.89%
+	Écart	10.58%	10.11%	10.49%	10.67%	11.77%	11.05%	11.21%	11.01%	10.62%	11.36%	
Localloc(2)	Tps CPU	356.58	348.94	404.68	387.16	392.46	391.24	360.2	344.78	343.96	342.34	367.23
Recuit	Coût	6485762	6394718	6752445	6465783	6642064	6511521	6492305	6426745	6308822	6510833	6.18%
+	Écart	5.34%	5.87%	6.17%	5.95%	6.64%	6.28%	6.24%	6.32%	6.13%	6.87%	
Localloc(mf)	Tps CPU	289.84	254.8	301.58	277.88	297.12	280.32	277.1	267.08	264.36	280.26	279.03
Méthode	Coût	6280964	6176031	6501409	6265869	6378123	6277460	6252022	6177605	6108380	6254716	2.39%
hybride 1	Écart	2.02%	2.25%	2.22%	2.67%	2.40%	2.46%	2.31%	2.20%	2.76%	2.66%	
	Tps CPU	205	205	205	205	205	205	205	205	205	205	205.00
Méthode	Coût	6245867	6138582	6479694	6231991	6334461	6224925	6217847	6164444	6045528	6180719	1.73%
hybride 2	Écart	1.45%	1.63%	1.88%	2.12%	1.70%	1.60%	1.75%	1.98%	1.70%	1.45%	
	Tps CPU	2925	2920	3169	3013	3018	2903	2934	2908	2813	2811	2941.40

TABLE 5.5 – Résultats des différentes méthodes de résolution sur 10 instances avec en moyenne 2950 clients

a. Le branch & bound a atteint sa limite de 10800 secondes.

de localisation simple est limité à 10800 secondes (le temps de calcul peut-être supérieur à cause du premier noeud). Les solutions où le problème de localisation simple associé n'est pas résolu à l'optimum sont indiquées (dans ce cas, le branch & bound fournit une solution approchée). L'heuristique multi-phase est la méthode de résolution fournissant la meilleure solution sur l'ensemble des instances testées. Toutefois, on peut constater dans les tableaux 5.4 et 5.5 que sur les instances de très grande taille, le temps de calcul nécessaire à la résolution du problème de localisation simple de la phase 1 peut poser problème. Une solution pourrait être de résoudre ce problème à l'aide d'heuristiques plus rapides. Quoiqu'il en soit cela nous a amené à chercher un compromis entre qualité et temps de calcul parmi les autres méthodes présentées.

Trois différentes variantes du recuit simulé sont présentées. La première utilise uniquement une itération de LOCALLOC au moment de l'initialisation. La deuxième utilise deux itérations de LOCALLOC à chaque palier de température. La dernière utilise LOCALLOC jusqu'à l'obtention d'un minimum local à chaque palier de température.

Pour les heuristiques hybrides, nous utilisons 500 réalisations du processus ponctuel représentant les sous-répartiteurs.

La figure 5.8 reprend les points clefs des résultats présentés dans les tableaux. Au niveau de la qualité des solutions, il apparaît que les heuristiques hybrides utilisant la géométrie stochastique s'en sortent beaucoup mieux que le recuit simulé, et résistent très bien à l'augmentation de la taille des instances. À ce titre, la seconde heuristique hybride (utilisant LOCALLOC sur l'ensemble des tirages stochastiques) est celle qui fournit les solutions les plus proches de celles de l'heuristique multi-phase. Mais la première heuristique hybride (utilisation de LOCALLOC uniquement sur le meilleur tirage) s'en rapproche au fur et à mesure de l'augmentation de la taille des instances.

En revanche, au niveau du temps de calcul, la figure 5.8 nous montre que ce sont la première heuristique hybride et les méthodes de recuit simulé utilisant LOCALLOC qui résistent le mieux. La seconde heuristique montre une augmentation linéaire de son temps de calcul en fonction de la taille des instances, ce qui peut devenir problématique. Toutefois nous rappelons que ce temps de calcul est directement lié au nombre de tirages stochastiques utilisés. Ce paramètre pourrait donc être ajusté par l'utilisateur pour obtenir un compromis plus acceptable entre qualité et temps de calcul.

Comparons maintenant le nombre moyen de SR trouvé par l'ensemble des méthodes avec la figure 5.9. On retrouve dans cette figure des résultats similaires à ceux concernant la qualité des solutions. Les méthodes s'approchant le plus du nombre de SR obtenu dans l'heuristique multi-phase sont les deux heuristiques hybrides, qui l'encadrent littéralement.

Il ressort de ces tests que lorsque les instances sont de très grande taille, les heu-

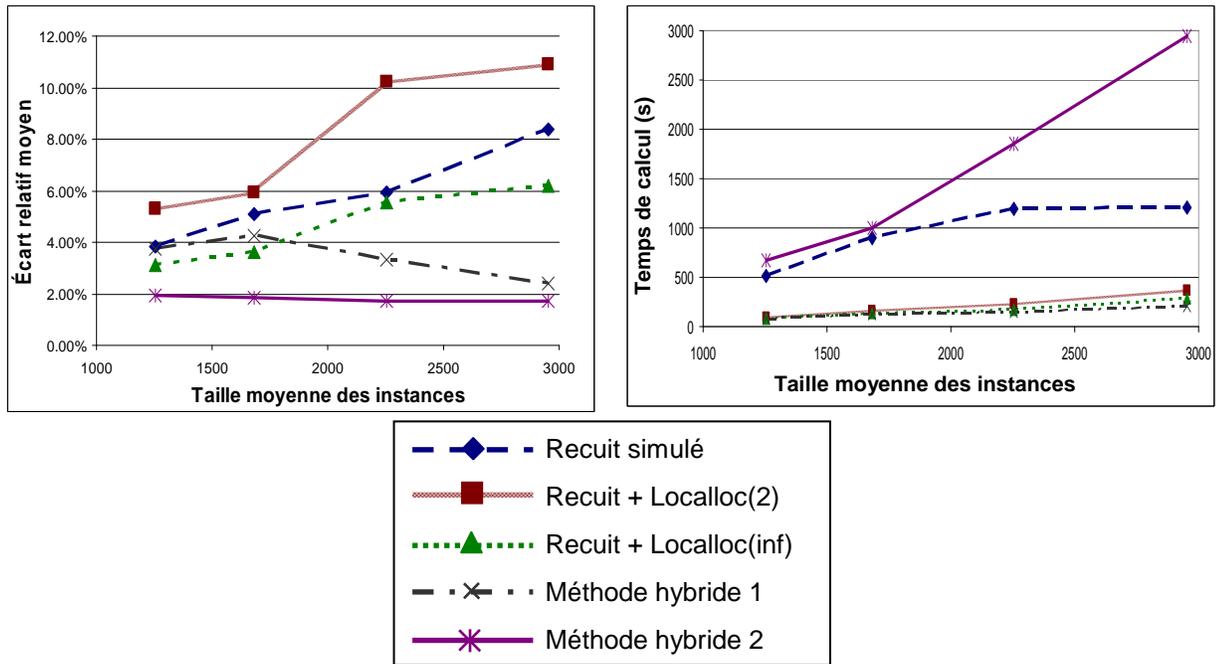


FIGURE 5.8 – Évolution de l'écart avec l'heuristique multi-phase et du temps de calcul selon la taille des instances

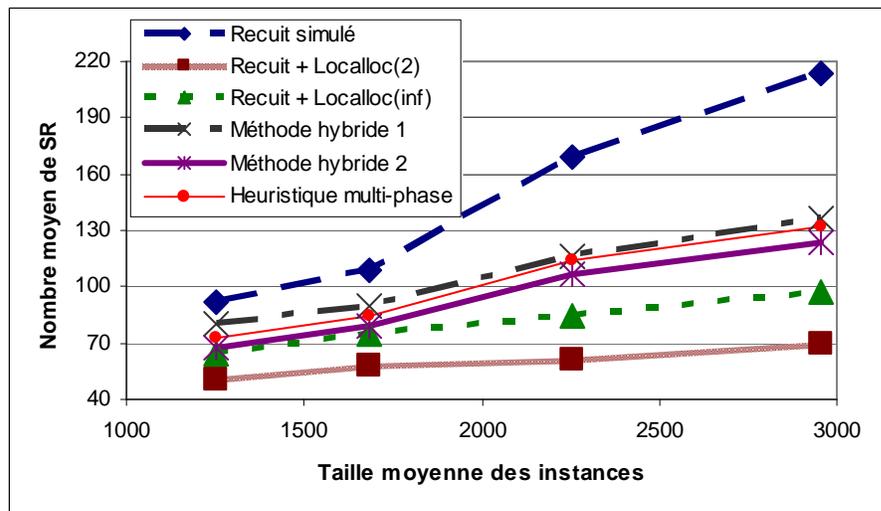


FIGURE 5.9 – Comparaison du nombre moyen de SR dans les solutions

ristiques hybrides alliant géométrie stochastique et LOCALLOC fournissent un bon compromis entre qualité des solutions et temps de calcul. Selon le temps dont on dispose et la taille des instances, on pourra choisir la première heuristique (en cas de données de très grande taille ou de fortes contraintes sur le temps de calcul), ou la seconde.

5.3.7 Expérimentations sur instances réelles

Dans la section précédente, les heuristiques hybrides ont été testées sur des instances générées à l'aide de la géométrie stochastique. On peut donc se poser la question de leur efficacité sur des instances réelles. Nous comparons donc ici l'heuristique multi-phase et les heuristiques hybrides sur deux instances réelles.

5.3.7.1 Adaptation de la méthodologie

Ici nous ne disposons en entrée que des coordonnées des clients et du NRA, ainsi que des paramètres de coûts. Or, pour les heuristiques hybrides, nous avons besoin de la densité de clients et de la densité de NRA afin de calculer la densité optimale de SR. Pour obtenir ces densités, nous allons utiliser l'idée introduite dans la section 4.1.2, à savoir calculer l'enveloppe convexe de l'ensemble des clients. Cette enveloppe convexe nous fournit la surface de l'instance et nous pouvons alors calculer les densité de clients et de NRA. L'intensité optimale du processus modélisant les SR est obtenue par formule et les réalisations du processus ont lieu à l'intérieur de l'enveloppe convexe des clients. Les solutions des trois méthodes sont alors comparées.

Le calcul de la surface de l'instance affecte le calcul des densités et donc le nombre de SR présents dans les solutions issues des heuristiques hybrides. Les clients des instances réelles ne sont pas répartis de façon homogène, parfois même certaines données peuvent être erronées, rendant l'enveloppe convexe mauvaise pour estimer la surface de l'instance. Afin d'étudier cet impact nous avons effectué des tests où quelques clients trop éloignés sont exclus du calcul de la surface de l'instance.

5.3.7.2 Résultats numériques

Nous présentons ici les résultats de l'heuristique multi-phase et des heuristiques hybrides sur deux instances issues de données réelles : MNA et DAN. Ces deux instances représentent plus de dix mille clients mais du fait de leur densité, il est possible de les réduire respectivement à 1761 et 2965 clients auxquels sont affectés des poids.

Le tableau 5.6 donne pour chacune des deux instances et pour chaque solution son coût et le temps de calcul nécessaire pour l'obtenir. Pour les heuristiques hybrides

	MNA			DAN		
	Solution	Temps CPU	Écart relatif	Solution	Temps CPU	Écart relatif
Heuristique multi-phase	8718622	1551		11526092	19196	
Heuristique hybride 1	9006729	68	3.30%	12237340	184	6.17%
Heuristique hybride 2	8998141	1306	3.21%	11984180	3437	3.97%

TABLE 5.6 – Résultats sur deux instances issues de données réelles

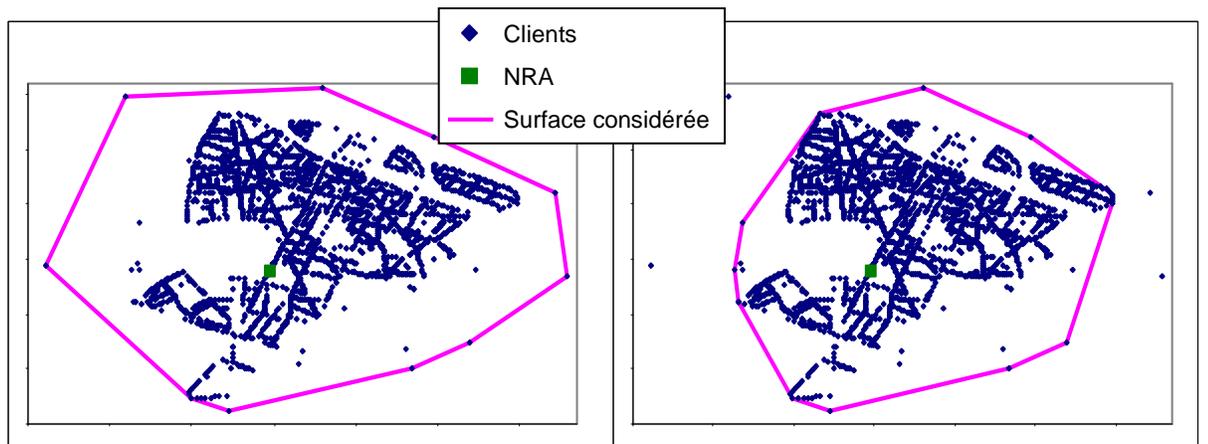


FIGURE 5.10 – Ajustement de la surface considérée pour le calcul des densités sur DAN

est également précisé l'écart relatif avec la valeur de la solution de l'heuristique multi-phase.

L'heuristique multi-phase est toujours efficace sur l'instance MNA, où son temps de calcul est d'un peu moins de 26 minutes. Il y a alors peu d'intérêt à utiliser une heuristique hybride. En revanche, la taille plus importante de DAN fait exploser le temps de calcul de l'heuristique multi-phase, et il peut être alors intéressant d'utiliser la seconde heuristique hybride. On constate également que les heuristiques hybrides sont un peu moins performantes pour DAN. Pour vérifier si cela est dû à une mauvaise estimation de la densité des clients et du NRA, nous avons utilisé les heuristiques hybrides avec une enveloppe convexe « ajustée » ne prenant pas en compte 4 clients, comme illustré dans la figure 5.10.

Les résultats des heuristiques hybrides s'améliorent sensiblement puisque leur écart relatif avec la solution de l'heuristique multi-phase passe respectivement à 3,81% et

3,08%. Un pré-traitement pour mettre de côté les clients isolés pourrait donc être utile en cas d'utilisation des heuristiques hybrides sur des instances réelles.

Nous avons essayé d'utiliser la génération de colonnes centrale sur MNA pour obtenir une borne inférieure. Le temps de calcul était malheureusement beaucoup trop long, et nous avons dû stopper la résolution. La dernière borne inférieure obtenue indique que la solution de l'heuristique multi-phase se trouve à moins de 10% de la valeur optimale.

5.4 Conclusion

Dans ce chapitre, nous avons développé un éventail de méthodes appropriées pour résoudre un problème de localisation d'équipements réseau intermédiaires sur des instances de grandes tailles. Selon la taille des données et le temps de calcul disponible, l'utilisation à la fois de l'optimisation combinatoire et de la géométrie stochastique fournit une solution adéquate.

Pour les instances les moins grandes, quand le temps de calcul n'est pas un problème, l'heuristique multi-phase adaptée de celle de [7] est celle qui fournit les solutions de meilleure qualité. La qualité des solutions a été vérifiée sur des instances de taille moyenne à l'aide de bornes inférieures obtenues par une nouvelle méthode de génération de colonnes centrale stabilisée. Pour les besoins de cette approche de génération de colonnes, nous avons résolu un nouveau sous-problème que nous avons appelé : problème de Weber avec coût de raccordement maximal.

Au fur et à mesure que la taille des instances augmente, ou que le temps de calcul disponible diminue, les heuristiques hybrides utilisant la modélisation du problème à l'aide d'outils de géométrie stochastique se révèlent être de très bons compromis, même sur les instances issues de données réelles.

Toutefois, en regardant la répartition des clients dans les instances réelles (voir figure 5.10), on s'aperçoit qu'ils ne sont pas répartis de façon homogène et l'on peut même deviner une voirie ou un réseau sous-jacent. Nous avons donc cherché à prendre en compte cette voirie ou ce réseau existant dans les problématiques de localisation d'équipements, ce qui nous a également amené à changer d'application pour passer au réseau en fibre optique.

Ce second cas d'application avec prise en compte d'une infrastructure existante est l'objet du chapitre suivant.

Chapitre 6

Déploiement d'un réseau d'accès optique FTTH¹ en présence d'infrastructures existantes

6.1 Introduction

Dans le chapitre précédent la localisation des équipements s'effectuait dans le plan. Ces modèles permettent des études à grande, et même très grande échelle. Mais les hypothèses effectuées ne correspondent pas toujours à la réalité opérationnelle. À l'issue de la section 5.3.7, nous avons noté la non-homogénéité de la distribution des clients et l'existence d'une structure sous-jacente. Dans le cas des zones urbaines notamment, les câbles ne peuvent passer partout. Ils suivent le plus souvent le réseau de voirie. Une infrastructure (génie civil ou réseau existant) est également souvent déjà présente sur la zone. Cette infrastructure influence non seulement le raccordement des équipements entre eux mais également leur placement qui peut être restreint à des sites bien précis. Nous voulions donc en tenir compte dans nos modèles.

Ceci nous a amené à changer d'application pour nous tourner vers le déploiement de réseau optique FTTH. En effet, la FTTH est un nouveau réseau et les investissements nécessaire pour le déployer sont énormes, notamment les coûts de génie civil. Pour cette raison, France Télécom cherche au maximum à réutiliser son génie civil existant (*i.e.* les chambres et conduites déjà installées pour d'autres réseaux et où il reste de la place pour faire passer des câbles optiques) et le nouveau génie civil devra suivre la voirie. Le déploiement de réseau FTTH est donc une application où l'infrastructure existante a une importance cruciale.

1. *Fiber To The Home* : désigne une architecture de réseau optique amenant la fibre optique jusqu'au logement de l'abonné.

Dans ce chapitre, nous présenterons d'abord les différentes architectures des réseaux FTTH. Ensuite, nous expliquerons les enjeux et les problématiques de conception de réseaux rencontrées par France Télécom. La réalité opérationnelle étant complexe, nous avons séparé le problème de conception de réseau FTTH en deux problèmes : le problème de localisation des coupleurs optiques (les équipements intermédiaire entre l'équipement central et les clients), et le problème d'affectation de câblage. Ces deux problèmes sont présentés dans des sections séparées. Enfin, nous présenterons des résultats de tests effectués sur deux instances réelles où nos modèles sont utilisés pour comparer deux choix d'architectures.

6.1.1 Le réseau d'accès FTTH

L'utilisation des réseaux de télécommunications IP pour des usages de plus en plus gourmands en bande passante (transfert de fichiers, *streaming* vidéo ou audio, *cloud computing*, ...), ainsi que leur fusion progressive avec les réseaux de téléphonie fixe et mobile, augmente le débit nécessaire aux utilisateurs. Pour répondre à ces besoins, les opérateurs de télécommunications ont recours à la fibre optique pour remplacer l'actuelle paire de cuivre. Pour permettre des débits toujours plus élevés la fibre descend dans le réseau, de plus en plus près de l'abonné. Dans la figure 6.1, on voit que certaines architectures apportent la fibre aux environs du clients (FTTN² ou FTTC³), d'autres vont plus loin jusqu'au bâtiment (FTTB), et enfin la FTTH va jusqu'au logement de l'abonné.

La FTTH est le choix le plus pérenne en terme d'évolution du débit car le client est raccordé intégralement en fibre optique. Mais pour la même raison, c'est aussi le choix nécessitant le plus d'investissements, car on ne réutilise pas du tout la paire de cuivre du réseau téléphonique commuté.

Deux architectures fth sont possibles pour relier les abonnés à l'équipement central optique (nommé OLT⁴) :

Point à Point : chaque abonné est connecté à une fibre optique directement reliée à l'équipement central.

Point à multi-point : une fibre sortant de l'OLT passe par un ou plusieurs coupleurs optiques passifs qui distribuent le débit d'une fibre en entrée sur plusieurs fibres en sortie. Ainsi, un nombre de clients dépendant de la capacité des coupleurs peuvent être connectés à l'OLT par une même fibre.

France Télécom a fait le choix d'utiliser une architecture point à multi-point suivant la norme GPON⁵ avec un taux de partage de 1 : 64 (*i.e.* jusqu'à 64 clients peuvent être

2. *Fiber To The Neighbourhood* : la fibre va jusqu'au voisinage du client.

3. *Fiber To The Curb* : littéralement, la fibre va jusqu'au trottoir du client.

4. *Optical Line Termination* : Désigne l'équipement central optique agrégeant le trafic de l'ensemble des clients d'une zone.

5. *Gigabit capable Passive Network Architecture*

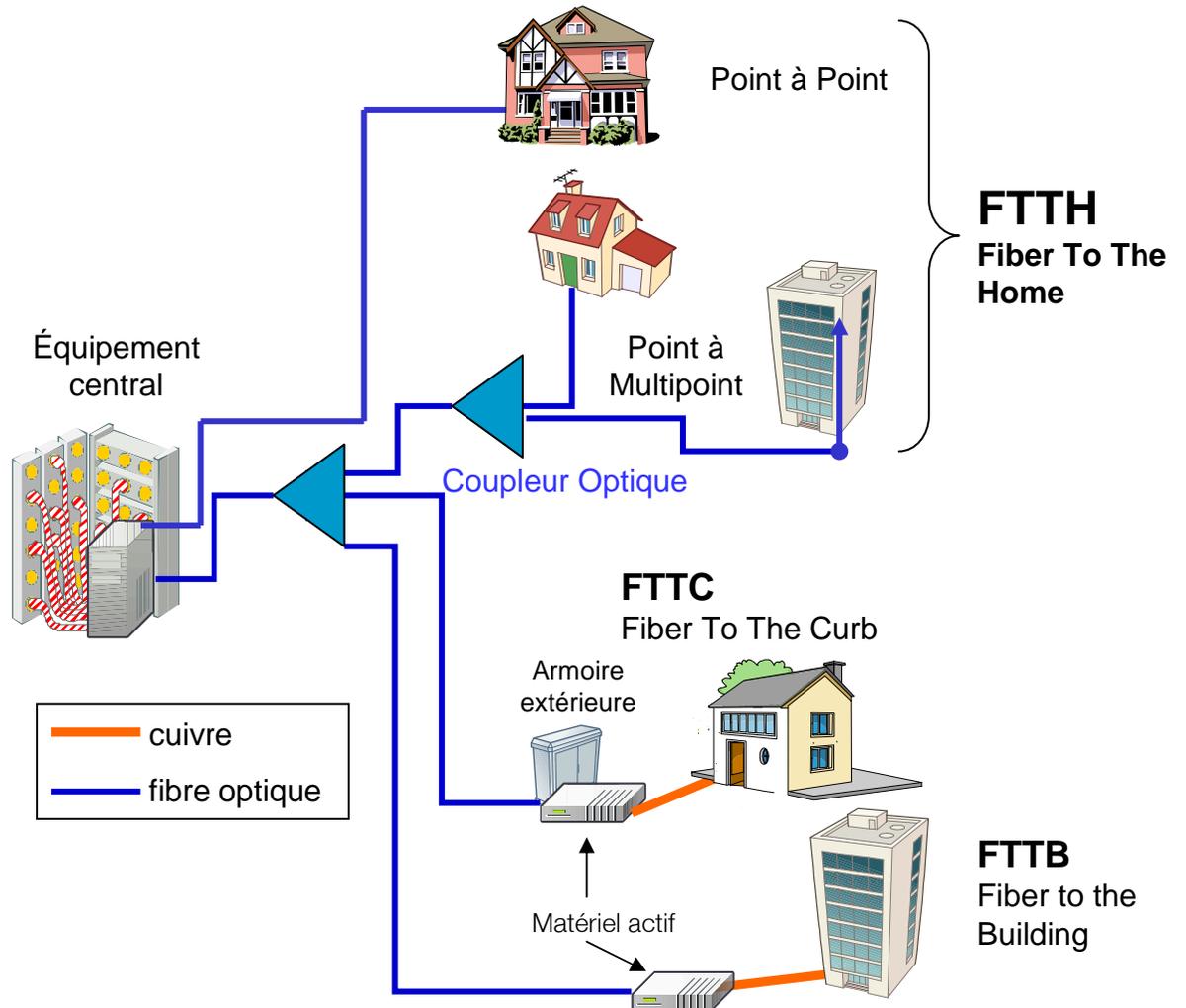


FIGURE 6.1 – Les différentes architectures FTTx

connectés à une même fibre sortant de l'OLT). Dans les pilotes actuels de déploiement FTTH, ce taux de partage est atteint en utilisant deux niveaux de coupleurs optiques passifs 1 : 8 et nous avons un réseau hiérarchique à plusieurs niveaux :

1. Le NRO (Noeud de Raccordement Optique ou NORA : Noeud Optique de Raccordement des Abonnés) est le lieu où est installé l'OLT.
2. Les PDZ (Points de Distribution de Zone) sont les endroits où sont installés le premier niveau de coupleurs alimentés par des fibres venant de l'OLT.
3. Les PE (Points d'éclatement) sont les endroits où sont regroupés les câbles optiques d'un voisinage. Selon les règles d'ingénierie, des coupleurs de deuxième niveau peuvent être installés dans ces endroits.
4. Les PRI (Point de Raccordement d'Immeuble ou Point de Raccordement d'Îlot) sont les premiers point d'agrégation juste avant les abonnés. Ils sont soit placés au pied des immeubles, soit à proximité de plusieurs habitations en cas de logements individuels. Selon les règles d'ingénierie, des coupleurs de deuxième niveau peuvent être installés dans ces endroits.
5. Les logements des abonnés.

Ces éléments sont représentés dans la figure 6.2. Dans la liste des éléments du réseau, on voit bien qu'à la fois les emplacements des coupleurs mais également le choix du câblage, sont importants.

6.1.2 Problématiques et enjeux pour France Télécom

Le choix de la FTTH entraîne le déploiement d'un tout nouveau réseau d'accès. Aucune partie des paires de cuivre du Réseau Téléphonique Commuté, utilisées à l'heure actuelle pour le haut débit internet (adsl), n'est réutilisée. Les investissements sont donc énormes, en particulier les coûts de génie civil.

Pour économiser grandement sur le génie civil, il est possible de réutiliser le génie civil existant, c'est à dire les conduites déjà posées pour d'autres réseaux et où il reste de la place pour faire passer des câbles optiques. C'est particulièrement le cas en zone urbaine où France Télécom dispose d'un génie civil existant abondant. Par conséquent, France Télécom ne veut poser de nouvelles conduites qu'en dernier recours sur ces zones. Pour les zones moins denses, le génie civil existant peut ne pas suffire et il faut donc l'intégrer à la problématique de déploiement. Mais même dans ce cas les nouvelles conduites ne pourront passer n'importe où et devront le plus souvent suivre la voirie.

Du fait du génie civil, l'infrastructure existante prend énormément d'importance dans le déploiement du réseau d'accès FTTH. Il est donc nécessaire de l'intégrer à nos modèles. Nous ne pouvons plus effectuer une localisation des équipements n'importe où dans le plan comme dans le chapitre précédent.

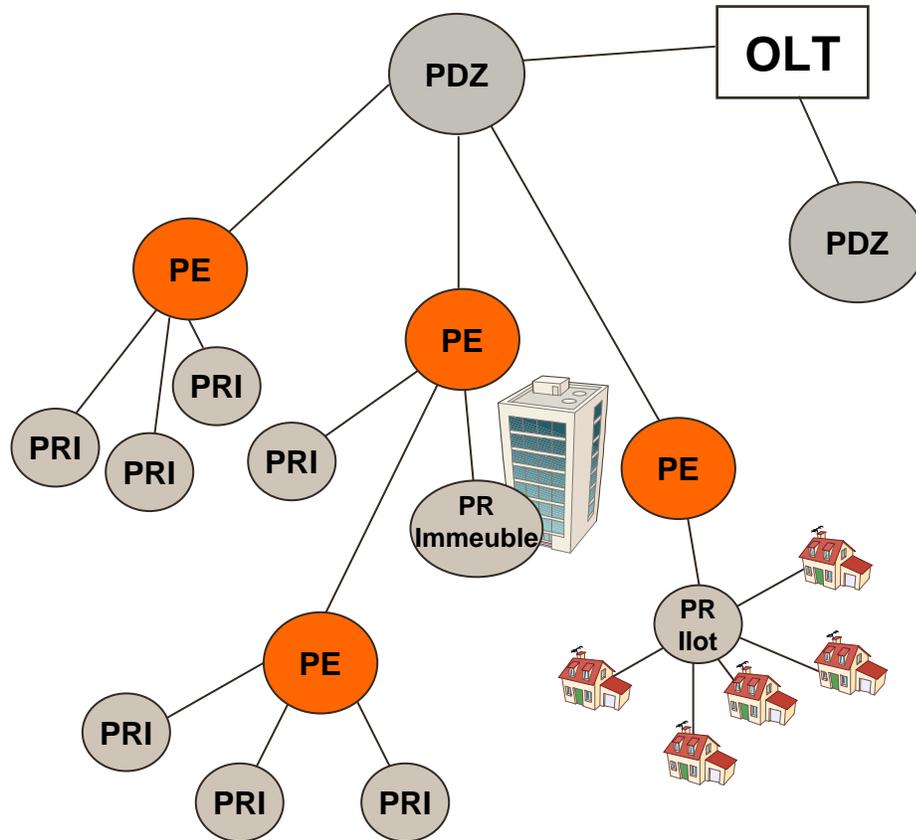


FIGURE 6.2 – Les éléments de réseau d'accès FTTH dans l'architecture actuelle

Actuellement, les pilotes de déploiement FTTH s'appuient sur un ensemble de règles d'ingénierie censées fournir un cadre de « bonnes pratiques » menant à un déploiement efficace. Ces règles portent sur la localisation des coupleurs, le cadre d'utilisation du génie civil existant, le dimensionnement des noeuds du réseau où sont installés les coupleurs, le choix des câbles et leur raccordement, etc... Certaines correspondent à des contraintes techniques, d'autres à des choix d'architecture. Elles sont issues des expérimentations sur le terrain. Elles doivent répondre au principaux enjeux pour France Télécom :

- La réutilisation du génie civil existant ;
- Le choix d'une architecture pour le nombre, la localisation, et la capacité des coupleurs optiques ;
- Le choix d'un plan de câblage pour raccorder les différents éléments du réseau entre eux ;
- Accroître le taux d'éligibilité (le pourcentage de logements pouvant accéder à la fibre optique par rapport à l'ensemble des logements de la zone.

Les modèles mathématiques que nous allons présenter dans ce chapitre ont pour but d'aider à l'évaluation des règles d'ingénierie et des architectures en les comparant à une solution « idéale » issue de la résolution de nos modèles. Pour fournir de bons éléments de comparaison, nos modèles doivent donc s'approcher le plus possible des exigences opérationnelles.

La taille des données étant importante, il est trop compliqué de prendre en compte l'ensemble des enjeux dans un même modèle. Dans [35], les auteurs présentent une heuristique pour optimiser à la fois la localisation de trois niveaux d'équipements : des points de distribution (correspondant à nos PE), un seul niveau de coupleurs optiques, et des points d'éclatement (correspondant hiérarchiquement à nos PDZ) où peuvent s'effectuer des raccords de câbles. Leur modèle ne prend pas en compte d'infrastructure existante. L'heuristique se décompose en plusieurs phases. Elle localise d'abord les coupleurs, puis les points d'éclatements, et enfin les points de distribution.

Dans [25], les auteurs proposent un modèle d'optimisation de réseau FTTH localisant dans le plan et dimensionnant un niveau de coupleurs. Le modèle présenté s'apparente au problème de Weber multi-source avec des contraintes supplémentaires sur la longueur maximale des liens et sur la capacité des équipements. Les auteurs proposent une heuristique alternant localisation et allocation sur le principe de LOCALLOC (voir section 4.2).

Dans [24], les auteurs proposent un modèle de programmation linéaire en nombre entiers pour optimiser le placement des coupleurs et le dimensionnement des câbles. Un seul niveau de coupleurs est pris en compte et le graphe sous-jacent est un arbre. Pour notre part, nous avons séparé la problématique de conception de réseau FTTH en deux parties :

- Le problème de localisation de coupleurs optiques :
 Dans ce problème, il s’agit de trouver le nombre et les emplacements des coupleurs optiques, ainsi que le raccordement des différents éléments par des fibres optiques (et éventuellement où rajouter de nouvelles conduites). Le but est de minimiser les coûts d’installation et de réutiliser le génie civil existant. Deux simplifications sont faites : la problématique du câblage n’est pas considérée (*i.e.* les fibres sont installées de façon individuelles), ni la problématique de l’éligibilité (*i.e.* on considère que la cible de clients à raccorder a été prouvée éligible par un autre moyen).
 - Le problème d’affectation de câblage :
 Ce problème prend en entrée la solution du problème de localisation de coupleurs optiques et détermine le câblage de moindre coût en prenant en compte l’impact des épissures ou soudures (connections entre deux câbles) sur l’éligibilité.
- Ces deux problèmes sont présentés dans les sections suivantes. Nous présenterons ensuite l’évaluation de ces modèles sur deux instances réelles.

6.2 Problème de localisation de coupleurs optiques

La partie accès d’un réseau FTTH, située entre l’OLT⁶ et les PRI⁷, se présente sous la forme d’un réseau hiérarchique. Entre les deux, nous trouvons deux ou trois niveaux de coupleurs optiques passifs (*i.e.* non alimenté électriquement), selon l’architecture choisie. Il y a plusieurs types de coupleurs qui démultiplient une fibre en 2, 4, 8, ou 16 fibres en sortie. Pour une fibre partant de l’OLT, nous voulons pouvoir satisfaire au maximum 64 clients. On dit que nous voulons avoir un taux de partage de 1 : 64

Une des particularités de notre problème est que nous disposons d’un graphe G représentant un réseau de génie civil existant (*i.e.* un ensemble de conduites et de chambres en béton dans lequel on peut faire passer les fibres et installer les coupleurs) et/ou une voirie, et sur lequel se trouvent l’OLT et les PRI. Chaque arête de ce graphe a une longueur et une capacité représentant le nombre maximum de fibres optiques pouvant transiter dans les conduites existantes le long de cette arête. Ces capacités empêchent de représenter un lien entre deux éléments du réseau par une seule arête directe. Ceci implique une topologie de réseau du type arbre-arbre-arbre (pour deux niveaux de coupleurs), et non arbre-étoile ou étoile-étoile comme c’est plus souvent le cas dans la littérature.

L’objectif de l’ensemble des modèles présentés dans cette section est de trouver les

6. *Optical Line Terminal* : concentrateur regroupant le trafic venant des clients pour le transmettre au coeur du réseau.

7. Point de Raccordement Immeuble/Îlot. Ordinairement placé au pied du bâtiment dont il agrège les demandes (PRImmeuble), il peut également servir à agréger de faibles demandes issues de plusieurs petits bâtiments (PRÎlot).

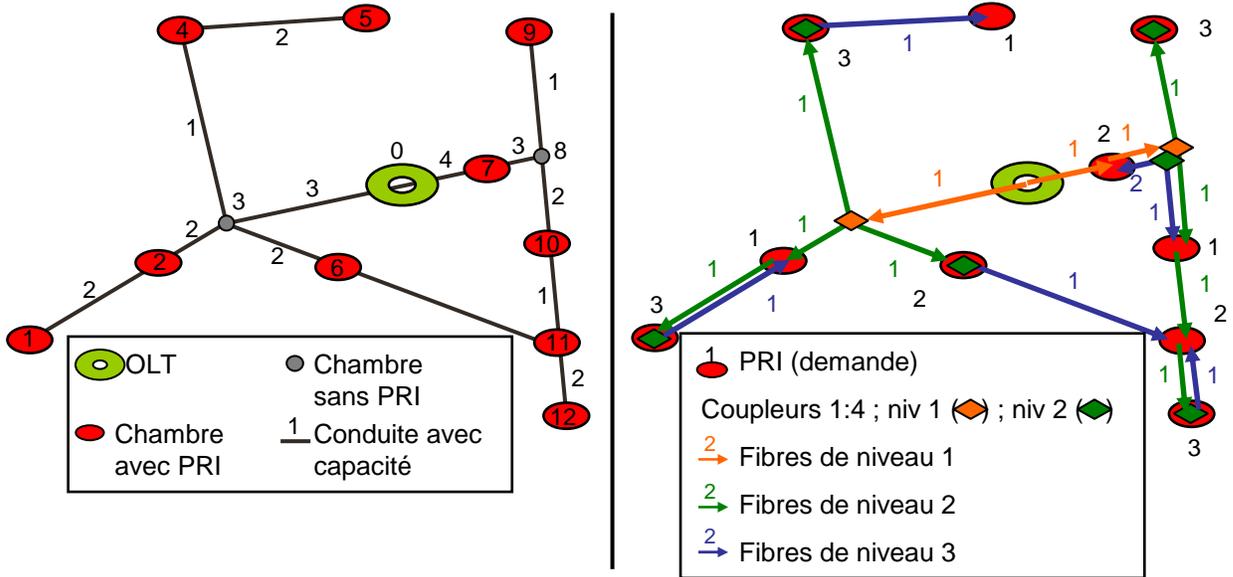


FIGURE 6.3 – Exemple d’instance avec une solution admissible

emplacements et le nombre de coupleurs optiques, ainsi que les fibres déployées, de façon à minimiser le coût des coupleurs et des fibres. Un exemple de génie civil existant et de solution de déploiement est présenté dans la figure . Il est également possible de considérer dans l’objectif le coût du nouveau génie civil si l’on autorise à augmenter la capacité des conduites existantes.

Dans un premier temps nous présenterons un modèle de base, où le nombre de fibres installées doit respecter la capacité existante (c-à-d sans nouveau génie civil), et où les PRI sont nécessairement raccordés à l’OLT en passant par deux coupleurs. Un ensemble d’inégalités valides est proposé pour ce modèle. Nous présentons ensuite une variante de ce modèle destinée à prendre en compte la volonté des opérationnels de dimensionner les coupleurs par rapport à une estimation du pourcentage d’abonnés. Enfin, nous présentons des modèles permettant la pose de nouvelles conduites et/ou les connections point à point (*i.e.* les PRI peuvent être connectés directement à l’OLT sans passer par des coupleurs).

6.2.1 Problèmes proches

Notre problème de localisations de coupleurs optiques est proche de nombreux problèmes d’optimisation dans les réseaux. En occultant certains aspects du problème, des problèmes classiques apparaissent.

Sans capacité sur les arêtes : Le graphe du génie civil existant peut alors être remplacé par un graphe complet dont les arêtes ont pour longueur celle du plus court chemin entre les deux sommets dans le graphe de génie civil. La topologie du réseau se transforme alors en étoile-étoile-étoile et le problème en variante du problème de localisation d'équipements avec capacités. Dans cette variante, les équipements doivent être dimensionnés (plusieurs coupleurs peuvent être placés sur le même site), et il y a deux niveaux d'équipements.

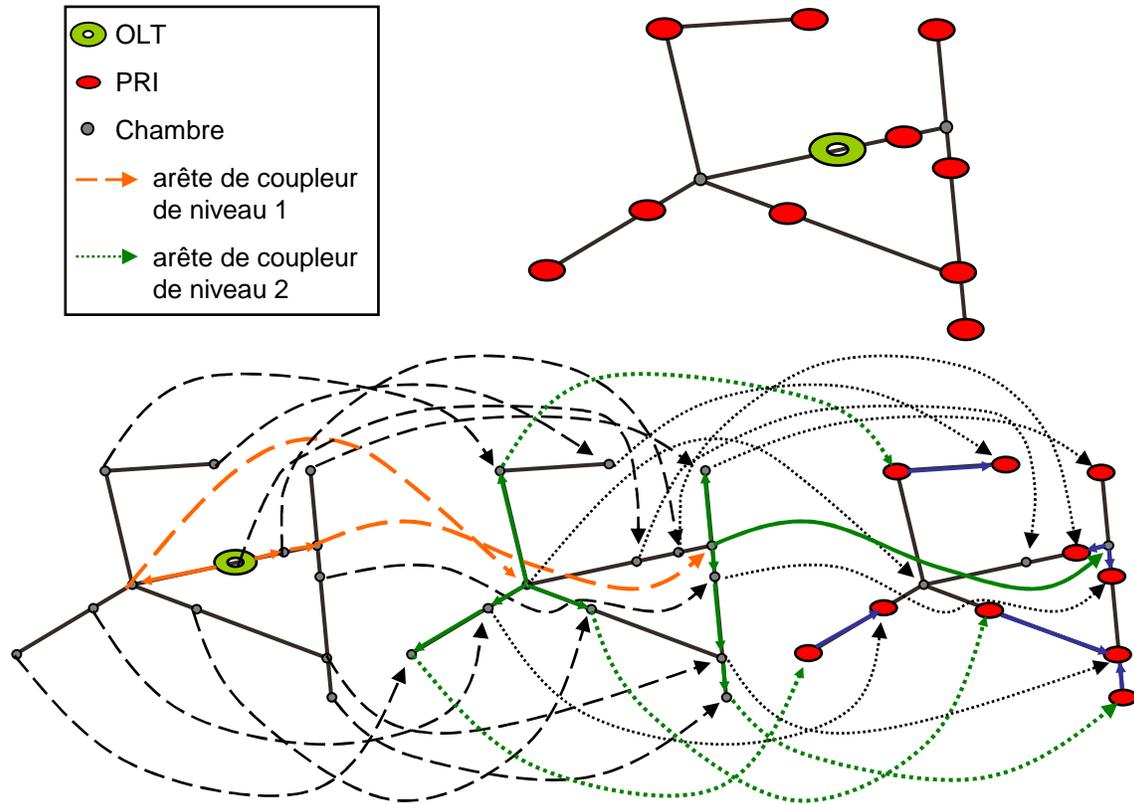
Sans coupleurs : Pour une architecture point-à-point où chaque client serait relié par une fibre individuelle, il n'y a plus de coupleurs à placer. Dans ce cas, on considère uniquement le routage des fibres et le creusement éventuel de nouvelles tranchées. On retombe alors sur des modèles de conception de réseaux d'accès classiques comme ceux présentés dans la section 3.1, par exemple dans [36].

Sans nouveau génie civil : Si l'on s'interdit la création de nouvelles tranchées, alors on peut rapprocher le problème d'un problème de flot généralisé entier dans un graphe $G'(V', E')$ obtenu à partir du graphe du génie civil existant $G(V, E)$. Pour obtenir G' , on prend trois copies orientées de G : $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, et $G_3 = (V_3, E_3)$. Chacune de ces copies correspond à un niveau de fibres optiques. Toutes les arêtes de ces copies ont la capacité de l'arête copiée et un multiplicateur de 1 (*i.e.* le flot de fibres entrant est égal au flot sortant). Notons $f_i : V \rightarrow V_i$, $i = 1, 2, 3$ les bijections qui associe à chaque élément de V sa copie dans V_i . Pour passer d'un niveau de fibres à l'autre, nous allons ajouter les ensembles d'arêtes suivantes : $E_{12} = \{(f_1(x), f_2(x)) \mid x \in V\}$ et $E_{23} = \{(f_2(x), f_3(x)) \mid x \in V\}$. Ces deux ensembles représentent les deux niveaux de coupleurs. Les arêtes de E_{12} (respectivement E_{23}) ont une capacité infinie et un multiplicateur correspondant à la capacité des coupleurs de niveau 1 (respectivement de niveau 2). Au final on a : $V' = V_1 \cup V_2 \cup V_3$, $E' = E_1 \cup E_2 \cup E_3 \cup E_{12} \cup E_{23}$. Sur la figure 6.4 est représenté le graphe G' correspondant à l'exemple de la figure 6.3.

Notre problème de localisation de coupleurs optiques généralise donc plusieurs problèmes difficiles rencontrés en optimisation des réseaux. C'est à la version sans nouveau génie civil que nous allons d'abord nous intéresser en proposant un premier modèle.

6.2.2 Modèle de base

Dans les zones denses, la pose de nouvelles conduites coûte extrêmement cher. On compte donc sur le génie civil existant pour faire passer la totalité des fibres optiques. Le génie civil existant est décrit de la façon suivante. Un ensemble de chambres en béton sont réparties sur la zone. Les chambres sont reliées entre elles par des tronçons

FIGURE 6.4 – Représentation du graphe G' dans le cas de l'exemple de la figure 6.3

artères. Chaque tronçon artère est constitué d'un ensemble d'alvéoles qui sont autant de conduites dans lesquelles peuvent passer des câbles optiques. Ces alvéoles sont parfois occupées par des câbles d'autres réseaux ou d'autres opérateurs. Leur capacité en nombre de fibres est donc variable.

Nous allons présenter une formulation s'appuyant sur trois niveaux de fibres, représentés chacun par des variables de flots. Les trois niveaux correspondent chacun à un niveau dans la hiérarchie du réseau :

1. le premier représente les fibres allant de l'OLT aux coupleurs de premier niveau ;
2. le second celles allant des coupleurs de premier niveau aux coupleurs de second niveau ;
3. le dernier représente les fibres issues des coupleurs de second niveau

Dans ce modèle, les coupleurs peuvent être placés uniquement dans les chambres et les demandes ne peuvent être satisfaites que par des fibres de niveau 3 (pas de connexion en point à point).

Soit $G(V, E)$ un graphe non orienté représentant le génie civil existant. Ses sommets sont indicés de 0 à n . Le sommet d'indice 0 est celui où se trouve l'OLT. Soit $\mathcal{C} \subseteq V$ l'ensemble des sommets correspondant aux PRI (ci-après dénommés clients). Soit $\mathcal{S} \subseteq V$ l'ensemble des sommets pouvant accueillir des coupleurs. On définit les variables et les notations suivantes :

- \mathbf{f}_{ij}^k $k = 1, 2, 3$: variables entières représentant le nombre de fibres optiques de niveau k empruntant l'arête $(i, j) \in E$ de i vers j ;
- \mathbf{z}_i^k $k = 1, 2$: variables entières indiquant le nombre de coupleurs de niveau k installés sur le sommet $i \in \mathcal{S}$;
- d_{ij} : la longueur en mètre de l'arête $(i, j) \in E$;
- C^k : le coût d'installation d'un coupleur de niveau k ;
- c^k : la capacité d'un coupleur de niveau k ;
- F_i : le coût au mètre d'une fibre optique de niveau i (ce prix par fibre est obtenu en divisant les prix des câbles prévus pour les différents niveaux par leur capacité) ;
- a_i : entier représentant la demande du sommet i (nulle si $i \notin \mathcal{C}$) ;
- b_{ij} : le nombre de fibres pouvant passer par l'arête (i, j) dans des tuyaux existants.

Le problème de localisation de coupleurs optiques se modélise sous la forme du programme linéaire en nombres entiers suivant :

$$\begin{array}{l}
\min_{\mathbf{F}, \mathbf{z}} \quad \sum_{i \in \mathcal{S}} (C^1 \mathbf{z}_i^1 + C^2 \mathbf{z}_i^2) + \sum_{(i,j) \in E} d_{ij} \cdot \sum_{k=1}^3 F^k (\mathbf{f}_{ij}^k + \mathbf{f}_{ji}^k) \\
\text{s.c. :} \\
\mathbf{z}_i^1 = \sum_{j \neq i} \mathbf{f}_{ji}^1 - \sum_{j \neq i} \mathbf{f}_{ij}^1 \quad \forall i \in \mathcal{S} \setminus \{0\} \quad (6.1) \\
\mathbf{z}_i^2 \leq \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 + c^1 \mathbf{z}_i^1 \quad \forall i \in \mathcal{S} \quad (6.2) \\
a_i \leq \sum_{j \neq i} \mathbf{f}_{ji}^3 - \sum_{j \neq i} \mathbf{f}_{ij}^3 + c^2 \mathbf{z}_i^2 \quad \forall i \in \mathcal{S} \quad (6.3) \\
0 = \sum_{j \neq i} \mathbf{f}_{ji}^1 - \sum_{j \neq i} \mathbf{f}_{ij}^1 \quad \forall i \in V \setminus (\mathcal{S} \cup \{0\}) \quad (6.4) \\
0 = \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 \quad \forall i \in V \setminus \mathcal{S} \quad (6.5) \\
a_i = \sum_{j \neq i} \mathbf{f}_{ji}^3 - \sum_{j \neq i} \mathbf{f}_{ij}^3 \quad \forall i \in V \setminus \mathcal{S} \quad (6.6) \\
\sum_{k=1}^3 (\mathbf{f}_{ij}^k + \mathbf{f}_{ji}^k) \leq b_{ij} \quad \forall (i, j) \in E \quad (6.7) \\
\mathbf{f}_{ij}^k, \mathbf{z}_i^k \in \mathbb{N}
\end{array}
\quad (\mathcal{LCO})$$

Les contraintes (6.1) et (6.4) assurent :

- que chaque coupleur de niveau 1 est alimenté par une fibre de niveau 1 ;
- la conservation du flot de niveau 1 lorsqu'aucun coupleur n'est installé sur le sommet i .

Les contraintes (6.2) et (6.5) reprennent le même principe pour les coupleurs de niveau 2 et le flot de niveau 2 en ajoutant les fibres de niveau 2 issues de coupleurs de niveau 1. Les contraintes (6.3) et (6.6) s'assurent que la demande en fibres du sommet i est satisfaite, ou que le flot de niveau 3 est conservé si la demande est nulle. Les contraintes (6.2) et (6.3) sont des inégalités car un coupleur peut être utilisé en dessous de sa capacité. Les contraintes (6.7) assurent qu'il y a suffisamment de place libre pour faire passer les fibres sur l'arête (i, j) .

Notons que dans ce modèle, il est tout à fait possible d'installer des coupleurs de niveau 2 sur le même sommet que des coupleurs de niveau 1, et d'alimenter les premiers avec des fibres directement issus des derniers. Notons également que l'ensemble de l'infrastructure est dimensionnée pour un taux de pénétration de 100% (*i.e.* en considérant que tous les clients potentiels souscriront à l'offre FTTH).

6.2.2.1 Inégalités valides basées sur les équations de flots

Dans cette section nous présentons des inégalités valides pour notre problème basées sur le principe des inégalités *mixed-integer rounding* (voir par exemple [46]).

Résultats préliminaires On considère S l'ensemble des points (\mathbf{x}, \mathbf{y}) satisfaisant :

$$\begin{cases} d\mathbf{x} + \mathbf{y} \geq a \\ \mathbf{x} \in \mathbb{N} \quad \mathbf{y} \in \mathbb{N} \end{cases}$$

avec $a \in \mathbb{N}, d \in \mathbb{N}$.

Le polyèdre P défini par les inégalités suivantes : $\begin{cases} d\mathbf{x} + \mathbf{y} \geq a \\ \mathbf{x} \geq 0 \quad \mathbf{y} \geq 0 \end{cases}$, contient S .

Considérons maintenant l'inégalité suivante :

$$r\mathbf{x} + \mathbf{y} \geq r(q+1) \tag{6.8}$$

avec r et q le reste et le quotient de la division entière de a par d : $a = q \times d + r$ avec $0 \leq r < d$.

Proposition : L'inégalité (6.8) est valide pour S .

Démonstration :

On a :

$$\begin{aligned} d\mathbf{x} + \mathbf{y} \geq a &\Leftrightarrow \mathbf{y} \geq (q - \mathbf{x}) \times d + r \\ \text{et (6.8)} &\Leftrightarrow \mathbf{y} \geq (q - \mathbf{x}) \times r + r \end{aligned}$$

\mathbf{x} étant entier, il se produit de deux chose l'une :

– soit $\mathbf{x} \leq q$, et on a alors :

$$\left. \begin{array}{l} \mathbf{y} \geq (q - \mathbf{x}) \times d + r \\ r < d \\ q - \mathbf{x} \geq 0 \end{array} \right\} \Rightarrow \mathbf{y} \geq (q - \mathbf{x}) \times r + r$$

et donc l'inégalité (6.8) est valide.

– soit $\mathbf{x} \geq q + 1$, et alors :

$$\left. \begin{array}{l} (q - \mathbf{x}) \times r + r \leq -r + r = 0 \\ \mathbf{y} \geq 0 \end{array} \right\} \Rightarrow (6.8)$$

□

On voit dans cette démonstration que l'inégalité proposée tronque le polyèdre P dans la région $q < \mathbf{x} < q + 1$, comme illustré dans la figure 6.5.

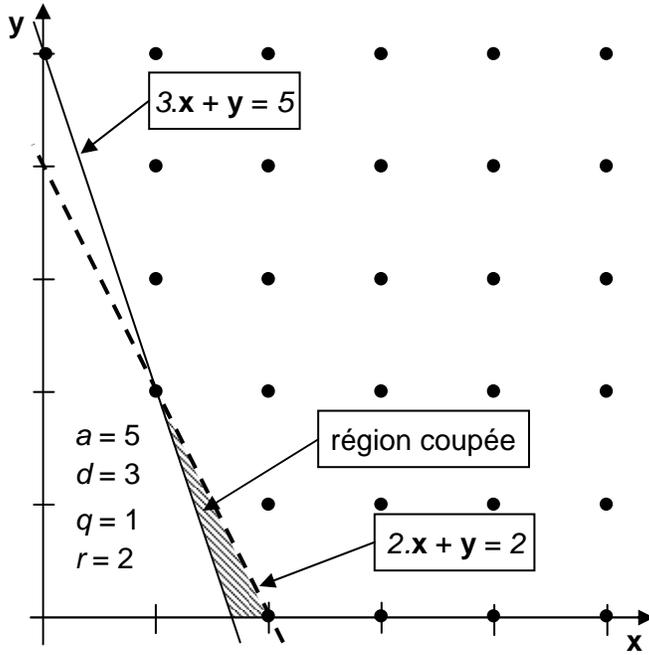


FIGURE 6.5 – Illustration d'une inégalité valide obtenue par rotation

Exemples d'application Soit A un ensemble de sommets $i \in V \setminus \{0\}$. Lorsqu'on additionne les équations (6.1) et (6.4) pour $i \in A$ les flots \mathbf{f}_{ij}^1 et \mathbf{f}_{ji}^1 pour $i, j \in A$ s'annulent. Il en va de même pour les contraintes des niveaux 2 et 3.

1. Pour le niveau 3, On obtient :

$$\begin{aligned} \sum_{i \in A} a_i &\leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 - \sum_{i \in A, j \notin A} \mathbf{f}_{ij}^3 + c^2 \sum_{i \in A \cap S} \mathbf{z}_i^2 \\ \Rightarrow \sum_{i \in A} a_i &\leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 + c^2 \sum_{i \in A \cap S} \mathbf{z}_i^2 \end{aligned} \quad (6.9)$$

En posant $\mathbf{x} = \sum_{i \in A \cap S} \mathbf{z}_i^2$, $\mathbf{y} = \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3$, et $a = \sum_{i \in A} a_i$ on obtient une inégalité de la forme $c^2 \mathbf{x} + \mathbf{y} \geq a$ avec $\mathbf{x}, \mathbf{y} \in \mathbb{N}$. On peut alors créer une inégalité valide selon la formule du paragraphe précédent :

$$\left(a - \left\lfloor \frac{a}{c^2} \right\rfloor c^2\right) \left\lceil \frac{a}{c^2} \right\rceil \leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 + \left(a - \left\lfloor \frac{a}{c^2} \right\rfloor c^2\right) \sum_{i \in A \cap S} \mathbf{z}_i^2 \quad (6.10)$$

2. En utilisant les inégalités de flots du niveau 2 $i \in A$, on a :

$$\begin{aligned}
\sum_{i \in A} a_i &\leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 + c^2 \sum_{i \in A \cap S} \mathbf{z}_i^2 \\
\Rightarrow \sum_{i \in A} a_i &\leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 + c^2 \left(\sum_{i \in A, j \notin A} \mathbf{f}_{ji}^2 - \sum_{i \in A, j \notin A} \mathbf{f}_{ij}^2 + c^1 \sum_{i \in A \cap S} \mathbf{z}_i^1 \right) \\
\Rightarrow \sum_{i \in A} a_i &\leq \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 + c^2 \mathbf{f}_{ji}^2) + c^2 c^1 \sum_{i \in A \cap S} \mathbf{z}_i^1 \quad (6.11)
\end{aligned}$$

En posant $\mathbf{x} = \sum_{i \in A \cap S} \mathbf{z}_i^1$, $\mathbf{y} = \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 + c^2 \mathbf{f}_{ji}^2)$, et $a = \sum_{i \in A} a_i$ on obtient une inégalité de la forme $c^2 c^1 \mathbf{x} + \mathbf{y} \geq a$ avec $\mathbf{x}, \mathbf{y} \in \mathbb{N}$. On peut alors créer une inégalité valide selon la formule du paragraphe précédent :

$$\left(a - \left\lfloor \frac{a}{c^2 c^1} \right\rfloor c^2 c^1 \right) \left\lceil \frac{a}{c^2 c^1} \right\rceil \leq \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 + c^2 \mathbf{f}_{ji}^2) + \left(a - \left\lfloor \frac{a}{c^2 c^1} \right\rfloor c^2 c^1 \right) \sum_{i \in A \cap S} \mathbf{z}_i^1 \quad (6.12)$$

3. On pourrait générer d'autres inégalités valides en combinant de façon différente les contraintes de flots.

6.2.3 Réduction du graphe sous-jacent

Le graphe représentant le génie civil existant et ou la voirie est un graphe de très grande taille (de mille à plusieurs milliers de sommets) mais très creux, même si ce n'est pas un arbre. Il contient donc de nombreux sommets de degré 2, parfois sans demande, dont un certain nombre peuvent être fusionnés car ils n'influent pas sur la solution. Des réductions du graphe peuvent donc être effectuées à la manière de celles pratiquées pour le problème de l'arbre de Steiner (voir [21]).

Propriété 6.1. *Si le sommet v de degré 1 n'a pas de demande ($a_v = 0$), alors dans la solution optimale du problème on a :*

$$f_{vu}^{k*} = f_{uv}^{k*} = 0, (v, u) \in E, \forall k = 1 \dots 3 \quad \text{et} \quad z_v^{k*} = 0, \forall k = 1 \dots 3$$

Démonstration

Supposons qu'on dispose d'une solution optimale pour laquelle on a :

$$\exists k \in [1; 3] \mid f_{vu}^{k*} \neq 0 \quad \text{ou} \quad \exists k \in [1; 3] \mid z_v^{k*} = 0$$

Construisons alors la solution (\hat{f}, \hat{z}) de la façon suivante :

$$\forall k = 1 \dots 3, \begin{cases} \hat{f}_{vu}^k = \hat{f}_{uv}^k = \hat{z}_v^k = 0 \\ \hat{z}_u^k = z_u^{k*} + z_v^{k*} \\ \hat{f}_{ij}^k = f_{ij}^{k*}, \hat{f}_{ji}^k = f_{ji}^{k*} & \forall (i, j) \in E, (i, j) \neq (v, u) \\ \hat{z}_i^k = z_i^{k*} & \forall i \neq u, i \neq v \end{cases}$$

Les seules contraintes affectées sont les contraintes de flots de chaque niveau pour les sommets u et v . Pour le sommet v , on a :

$$\begin{aligned} \hat{z}_v^1 &= 0 = 0 = \hat{f}_{uv}^1 - \hat{f}_{vu}^1 \\ \hat{z}_v^2 &= 0 \leq 0 = \hat{f}_{uv}^2 - \hat{f}_{vu}^2 + c^1 \hat{z}_v^1 \\ 0 &\leq 0 = \hat{f}_{uv}^3 - \hat{f}_{vu}^3 + c^2 \hat{z}_v^2 \end{aligned}$$

Les contraintes sont donc satisfaites pour v . Pour le sommet u , on a :

$$\begin{aligned} \hat{z}_u^1 &= z_u^{1*} + z_v^{1*} \\ \text{(avec la contrainte (6.1) pour } u) &= \sum_{j \neq u, v} f_{ju}^{1*} - \sum_{j \neq u, v} f_{uj}^{1*} + f_{vu}^{1*} - f_{uv}^{1*} + z_v^{1*} \\ \text{(avec la contrainte (6.1) pour } v) &= \sum_{j \neq u, v} \hat{f}_{ju}^1 - \sum_{j \neq u, v} \hat{f}_{uj}^1 + f_{vu}^{1*} - f_{uv}^{1*} + f_{uv}^{1*} - f_{vu}^{1*} \\ \text{(avec la définition de } \hat{f}_{ij}^1) &= \sum_{j \neq u} \hat{f}_{ju}^1 - \sum_{j \neq u} \hat{f}_{uj}^1 \\ \hat{z}_u^2 &= z_u^{2*} + z_v^{2*} \\ \text{(avec la contrainte (6.2) pour } u) &\leq \sum_{j \neq u, v} f_{ju}^{2*} - \sum_{j \neq u, v} f_{uj}^{2*} + f_{vu}^{2*} - f_{uv}^{2*} + c^1 z_u^{1*} + z_v^{2*} \\ \text{(avec la contrainte (6.2) pour } v) &\leq \sum_{j \neq u, v} \hat{f}_{ju}^2 - \sum_{j \neq u, v} \hat{f}_{uj}^2 + f_{vu}^{2*} - f_{uv}^{2*} + f_{uv}^{2*} - f_{vu}^{2*} + c^1 (z_u^{1*} + z_v^{1*}) \\ \text{(avec la définition de } \hat{f}_{ij}^2 \text{ et de } \hat{z}_u^1) &\leq \sum_{j \neq u} \hat{f}_{ju}^2 - \sum_{j \neq u} \hat{f}_{uj}^2 + c^1 \hat{z}_u^1 \\ \text{(contrainte (6.3) pour } u) \quad a_u &\leq \sum_{j \neq u, v} f_{ju}^{3*} - \sum_{j \neq u, v} f_{uj}^{3*} + f_{vu}^{3*} - f_{uv}^{3*} + c^2 z_u^{2*} \\ \text{(avec la contrainte (6.3) pour } v) &\leq \sum_{j \neq u, v} f_{ju}^{3*} - \sum_{j \neq u, v} f_{uj}^{3*} + f_{vu}^{3*} - f_{uv}^{3*} + c^2 z_u^{2*} + c^2 z_v^{2*} + f_{uv}^{3*} - f_{vu}^{3*} \\ \text{(avec la définition de } \hat{f}_{ij}^3 \text{ et de } \hat{z}_u^2) &\leq \sum_{j \neq u} \hat{f}_{ju}^3 - \sum_{j \neq u} \hat{f}_{uj}^3 + c^2 \hat{z}_u^2 \end{aligned}$$

La solution (\hat{f}, \hat{z}) respecte donc l'ensemble des contraintes du problème, tout en faisant l'économie des fibres installées sur l'arête (u, v) . La solution (f^*, z^*) n'est donc pas optimale. \square

La première phase de réduction consiste donc à retirer successivement les sommets de degré 1 sans demande du graphe G . Retirer ces sommets peut faire apparaître de nouveaux sommets de degré 1 qui sont retirés à leur tour.

Passons maintenant au sommets de degré 2 sans demande dont les arêtes incidentes ont la même capacité.

Propriété 6.2. *Si le sommet v de degré 2 n'a pas de demande ($a_v = 0$), et si ses deux arêtes incidente (u, v) et (v, w) ont la même capacité alors il existe une solution optimale du problème qui vérifie :*

$$z_v^{k*} = 0, \forall k = 1 \dots 3$$

Démonstration

Supposons que l'on dispose d'une solution optimale où :

$$\exists k = 1 \dots 3 \mid z_v^{k*} > 0$$

Supposons sans perte de généralité que cette solution vérifie :

$$\sum_{k=1}^3 F^k (f_{vu}^{k*} + f_{uv}^{k*}) \geq \sum_{k=1}^3 F^k (f_{uv}^{k*} + f_{vu}^{k*})$$

Considérons alors la solution (\hat{f}, \hat{z}) construite de la façon suivante :

$$\forall k = 1 \dots 3, \begin{cases} \hat{z}_v^k = 0 \\ \hat{z}_u^k = z_u^{k*} + z_v^{k*} \\ \hat{f}_{uv}^k = f_{vw}^{k*}, \hat{f}_{vu}^k = f_{uv}^{k*} & \text{(possible car les deux arêtes ont la même capacité)} \\ \hat{f}_{ij}^k = f_{ij}^{k*}, \hat{f}_{ji}^k = f_{ji}^{k*} & \forall (i, j) \in E, (i, j) \neq (v, u) \\ \hat{z}_i^k = z_i^{k*} & \forall i \neq u, i \neq v \end{cases}$$

La différence de coût entre la solution (\hat{f}, \hat{z}) et la solution optimale est négative :

$$\begin{aligned} & \sum_{k=1}^2 C^k (\hat{z}_k^k + \hat{z}_u^k - z_u^{k*} - z_v^{k*}) + \sum_{k=1}^3 F^k (\hat{f}_{uv}^k + \hat{f}_{vu}^k - f_{uv}^{k*} - f_{vu}^{k*}) \\ &= \sum_{k=1}^3 F^k (f_{vw}^{k*} + f_{uv}^{k*}) - \sum_{k=1}^3 F^k (f_{uv}^{k*} + f_{vu}^{k*}) \leq 0 \end{aligned}$$

La solution est donc meilleure. Vérifions maintenant qu'elle vérifie les contraintes. Les seules contraintes affectées sont celles des sommets u et v . La contraintes de capacité sur (u, v) est respectée puisque celle de (w, v) est respectée par la solution optimale. Considérons maintenant les contraintes de flots pour le sommet v :

$$\begin{aligned}
\hat{f}_{uv}^1 + \hat{f}_{vw}^1 - \hat{f}_{vu}^1 - \hat{f}_{vw}^1 &= f_{vw}^{1*} + f_{uv}^{1*} - f_{uv}^{1*} - f_{vw}^{1*} \\
&= 0 = \hat{z}_v^1 \\
\hat{f}_{uv}^2 + \hat{f}_{vw}^2 - \hat{f}_{vu}^2 - \hat{f}_{vw}^2 + c^1 \hat{z}_v^1 &= f_{vw}^{2*} + f_{uv}^{2*} - f_{uv}^{2*} - f_{vw}^{2*} \\
&= 0 \geq 0 = \hat{z}_v^2 \\
\hat{f}_{uv}^3 + \hat{f}_{vw}^3 - \hat{f}_{vu}^3 - \hat{f}_{vw}^3 + c^2 \hat{z}_v^2 &= f_{vw}^{3*} + f_{uv}^{3*} - f_{uv}^{3*} - f_{vw}^{3*} \\
&= 0 \geq 0 = a_v
\end{aligned}$$

Les contraintes de flots sont donc respectées pour le sommet v . Passons maintenant au sommet u :

$$\begin{aligned}
\sum_{i \neq u, i \neq v} \hat{f}_{iu}^1 - \sum_{i \neq u, i \neq v} \hat{f}_{ui}^1 + \hat{f}_{vu}^1 - \hat{f}_{uv}^1 &= \sum_{i \neq u, i \neq v} f_{iu}^{1*} - \sum_{i \neq u, i \neq v} f_{ui}^{1*} + f_{uv}^{1*} - f_{uv}^{1*} \\
&= \sum_{i \neq u} f_{iu}^{1*} - \sum_{i \neq u} f_{ui}^{1*} + f_{uv}^{1*} + f_{uv}^{1*} - f_{uv}^{1*} - f_{uv}^{1*} \\
&= z_u^{1*} + z_v^{1*} = \hat{z}_u^1 \\
\sum_{i \neq u, i \neq v} \hat{f}_{iu}^2 - \sum_{i \neq u, i \neq v} \hat{f}_{ui}^2 + \hat{f}_{vu}^2 - \hat{f}_{uv}^2 + c^1 \hat{z}_u^1 &= \sum_{i \neq u, i \neq v} f_{iu}^{2*} - \sum_{i \neq u, i \neq v} f_{ui}^{2*} + f_{uv}^{2*} - f_{uv}^{2*} + c^1 z_u^{1*} + c^1 z_v^{1*} \\
&= \sum_{i \neq u} f_{iu}^{2*} - \sum_{i \neq u} f_{ui}^{2*} + c^1 z_u^{1*} + f_{uv}^{2*} + f_{uv}^{2*} - f_{uv}^{2*} - f_{uv}^{2*} + c^1 z_v^{1*} \\
&\geq z_u^{2*} + z_v^{2*} = \hat{z}_u^2 \\
\sum_{i \neq u, i \neq v} \hat{f}_{iu}^3 - \sum_{i \neq u, i \neq v} \hat{f}_{ui}^3 + \hat{f}_{vu}^3 - \hat{f}_{uv}^3 + c^2 \hat{z}_u^2 &= \sum_{i \neq u, i \neq v} f_{iu}^{3*} - \sum_{i \neq u, i \neq v} f_{ui}^{3*} + f_{uv}^{3*} - f_{uv}^{3*} + c^2 z_u^{2*} + c^2 z_v^{2*} \\
&= \sum_{i \neq u} f_{iu}^{3*} - \sum_{i \neq u} f_{ui}^{3*} + c^2 z_u^{2*} + f_{uv}^{3*} + f_{uv}^{3*} - f_{uv}^{3*} - f_{uv}^{3*} + c^2 z_v^{2*} \\
&\geq a_u + a_v = a_u
\end{aligned}$$

La solution (\hat{f}, \hat{z}) respecte donc toutes les contraintes et présente un coût inférieur ou égal à la solution optimale.

□

La propriété (6.2) permet de réduire toute chaîne sans demande et de capacité uniforme à une seule arête.

Une heuristique de réduction a également été testée sur les graphes de voirie (plus grands que les graphes de génie civil existant). Elle consiste à éliminer tout sommet n'appartenant à aucun des plus courts chemins entre les sommets appartenant à l'ensemble $\{0\} \cup \mathcal{C}$ (clients et OLT). Cette réduction permet d'éliminer des parties du graphe qui sont sans demande et qui sont plus complexes que des chaînes. En revanche, elle ne garantit pas la conservation de la solution optimale.

Nous présentons ici d'autres propriétés intéressantes sur la solution optimale même si elles ne permettent pas une réduction du graphe.

Propriété 6.3. *Dans toute solution optimale, j'ai la propriété suivante :*

$$f_{ij}^{k*} \geq 1 \Rightarrow f_{ji}^{k*} = 0 \quad (6.13)$$

Démonstration.

Supposons qu'il existe i, j , et k tels que :

$$f_{ij}^{k*} \geq 1 \text{ et } f_{ji}^{k*} \geq 1$$

On peut supposer sans perte de généralité que $f_{ij}^{k*} \leq f_{ji}^{k*}$. Alors la solution de déploiement où l'on effectue le changement suivant :

$$f_{ij}^k = 0 \quad \text{et} \quad f_{ji}^k = f_{ji}^{k*} - f_{ij}^{k*}$$

respecte toutes les contraintes et présente un coût inférieur de $2F^k d_{ij} f_{ij}^{k*}$ à celui de la solution optimale. Il y a donc une contradiction. \square

De façon plus générale, il est possible de montrer qu'il n'y a pas de circuit de fibres dans la solution optimale.

Propriété 6.4. *Dans la solution optimale, il n'existe pas de séquence d'indice de sommets (v_1, \dots, v_p) telle que :*

$$f_{v_1 v_2}^{l*} \geq 1, f_{v_2 v_3}^{l*} \geq 1, \dots, f_{v_{p-1} v_p}^{l*} \geq 1, f_{v_p v_1}^{l*} \geq 1 \quad (6.14)$$

Démonstration

Supposons que l'on dispose d'une solution vérifiant (6.14). Par commodité notons $v_{p+1} = v_0 = v_1$. Sans perte de généralité, supposons que :

$$\min \left\{ \min_{i=1 \dots p} f_{v_i v_{i+1}}^{l*} \right\} = f_{v_1 v_2}^{l*}$$

Considérons la solution (\hat{f}, \hat{z}) suivante :

$$\begin{cases} \hat{z}_i^k = z_i^{k*} & \forall k = 1 \dots 3, \forall i \in V \\ \hat{f}_{v_i v_{i+1}}^l = f_{v_i v_{i+1}}^{l*} - f_{v_1 v_2}^{l*} & \forall i = 1 \dots p \\ \hat{f}_{v_1 v_2}^l = 0 \\ \hat{f}_{ij}^k = f_{ij}^{k*} & \forall k = 1 \dots 3, \forall (i, j) \in E \setminus \{(v_i, v_{i+1}) \mid i = 1 \dots p\} \end{cases}$$

Vérifions que cette solution satisfait les contraintes. Les seuls sommets affectés sont les sommets (v_1, \dots, v_p) . Pour les sommets $\{v_i, i = 1 \dots p\}$ on a :

$$\begin{aligned} \sum_{j \neq v_i} \hat{f}_{j v_i}^l - \sum_{j \neq v_i} \hat{f}_{v_i j}^l &= \sum_{j \neq v_i, j \neq v_{i-1}} f_{j v_i}^{l*} + \left(f_{v_{i-1} v_i}^{l*} - f_{v_1 v_2}^{l*} \right) - \sum_{j \neq v_i, j \neq v_{i+1}} f_{v_i j}^{l*} - \left(f_{v_i v_{i+1}}^{l*} - f_{v_1 v_2}^{l*} \right) \\ &= \sum_{j \neq v_i} f_{j v_i}^{l*} - \sum_{j \neq v_i} f_{v_i j}^{l*} \end{aligned}$$

Par conséquent, pour tout sommet $i \in V$:

$$\begin{aligned} \sum_{j \neq i} \hat{f}_{j i}^1 - \sum_{j \neq i} \hat{f}_{i j}^1 &= \sum_{j \neq i} f_{j i}^{1*} - \sum_{j \neq i} f_{i j}^{1*} \\ &= z_i^{1*} = \hat{z}_i^1 \\ \sum_{j \neq i} \hat{f}_{j i}^2 - \sum_{j \neq i} \hat{f}_{i j}^2 + c^1 \hat{z}_i^1 &= \sum_{j \neq i} f_{j i}^{2*} - \sum_{j \neq i} f_{i j}^{2*} + c^1 z_i^{1*} \\ &\geq z_i^{2*} = \hat{z}_i^2 \\ \sum_{j \neq i} \hat{f}_{j i}^3 - \sum_{j \neq i} \hat{f}_{i j}^3 + c^2 \hat{z}_i^2 &= \sum_{j \neq i} f_{j i}^{3*} - \sum_{j \neq i} f_{i j}^{3*} + c^2 z_i^{2*} \\ &\geq a_i \end{aligned}$$

Les contraintes de flots sont donc respectées par la solution (\hat{f}, \hat{z}) . Les contraintes de capacité sont également respectées puisque le nombre de fibres passant sur une arête dans la solution (\hat{f}, \hat{z}) est inférieur ou égal aux fibres passant sur l'arête dans la solution optimale. On a donc une solution respectant les contraintes et avec moins de fibres, donc un coût inférieur. \square

6.2.4 Dimensionnement des coupleurs

Les premières règles d'ingénierie prévoient de dimensionner les câbles à 100% mais pas les coupleurs. En effet, la fibre doit être amenée chez chaque client potentiel

mais seulement une partie d'entre eux souscriront à l'offre FTTH. S'il est compliqué de rajouter des câbles au fur et à mesure de l'augmentation du nombre de clients, il est facile de rajouter des coupleurs aux endroits où il y en a déjà, si les câbles sont déjà en place. Les coupleurs de niveau 2 seront donc dimensionnés pour un taux de pénétration cible donné, noté $cible_2$ et compris entre 0 et 1. C'est à dire que pour un immeuble avec une demande de 100, on installera des coupleurs que pour une demande de $\lceil 100 \times cible_2 \rceil$.

Pour les coupleurs de niveau 1, c'est plus complexe. Les coupleurs de niveau 1 sous-utilisés (si les souscriptions sont inférieures localement au taux de pénétration cible) induisent en pratique un surcoût au niveau de l'OLT. Tout comme les coupleurs de niveau 2, on veut donc réduire le nombre de coupleurs de niveau 1 installés mais, l'enjeu en terme de coût étant plus important, la réduction sera plus grande. Dans l'idéal, chaque coupleur de niveau 1 devrait être utilisé à plus de 50%. En l'absence de prévisions exactes sur la répartition des futurs souscripteurs, il est difficile de savoir combien installer de coupleurs et où. Nous ferons donc preuve de prudence en considérant un second taux de pénétration $cible_1$, inférieur au premier, qui sera utilisé pour dimensionner les coupleurs de niveau 1.

Pour tenir compte de ces règles dans nos modèles, on crée de nouvelles variables \mathbf{w}_i^1 et \mathbf{w}_i^2 correspondant au nombre de coupleurs qui seront effectivement installés sur le sommet i après dimensionnement. Avec ces variables, on modifie le modèle de la façon suivante :

$$\begin{array}{l}
\min_{\mathbf{F}, \mathbf{z}} \quad \sum_{i \in \mathcal{S}} (C^1 \mathbf{w}_i^1 + C^2 \mathbf{w}_i^2) + \sum_{(i,j) \in E} d_{ij} \cdot \sum_{k=1}^3 F^k(\mathbf{f}_{ij}^k + \mathbf{f}_{ji}^k) \\
\text{s.c. :} \\
\mathbf{z}_i^1 = \sum_{j \neq i} \mathbf{f}_{ji}^1 - \sum_{j \neq i} \mathbf{f}_{ij}^1 \quad \forall i \in \mathcal{S} \setminus \{0\} \quad (6.1) \\
\mathbf{z}_i^2 \leq \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 + c^1 \mathbf{z}_i^1 \quad \forall i \in \mathcal{S} \quad (6.2) \\
a_i \leq \sum_{j \neq i} \mathbf{f}_{ji}^3 - \sum_{j \neq i} \mathbf{f}_{ij}^3 + c^2 \mathbf{z}_i^2 \quad \forall i \in \mathcal{S} \quad (6.3) \\
0 = \sum_{j \neq i} \mathbf{f}_{ji}^1 - \sum_{j \neq i} \mathbf{f}_{ij}^1 \quad \forall i \in V \setminus (\mathcal{S} \cup \{0\}) \quad (6.4) \\
0 = \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 \quad \forall i \in V \setminus \mathcal{S} \quad (6.5) \\
a_i = \sum_{j \neq i} \mathbf{f}_{ji}^3 - \sum_{j \neq i} \mathbf{f}_{ij}^3 \quad \forall i \in V \setminus \mathcal{S} \quad (6.6) \\
0 \leq \frac{1}{cible_1} \mathbf{w}_i^1 - \mathbf{z}_i^1 \quad \forall i \in \mathcal{S} \quad (6.15) \\
0 \leq \frac{1}{cible_1} \mathbf{w}_i^2 - \mathbf{z}_i^2 \quad \forall i \in \mathcal{S} \quad (6.16) \\
\sum_{k=1}^3 (\mathbf{f}_{ij}^k + \mathbf{f}_{ji}^k) \leq b_{ij} \quad \forall (i, j) \in E \quad (6.7) \\
\mathbf{f}_{ij}^k, \mathbf{z}_i^k \in \mathbb{N}
\end{array}$$

(LCOdim)

Par rapport au modèle de base (\mathcal{LCO}), les variables \mathbf{z}_i^1 et \mathbf{z}_i^2 ont été remplacées dans l'objectif par les variables \mathbf{w}_i^1 et \mathbf{w}_i^2 . Les contraintes (6.15) et (6.16) ont également été rajoutées au modèle. Ces dernières imposent que le nombre de coupleurs de niveau k installé sur un noeud i soit égal à : $\mathbf{w}_i^k = \lceil cible_k \times \mathbf{z}_i^k \rceil$.

Ces contraintes de dimensionnement vont malheureusement dégrader la qualité de la relaxation continue. En effet, en continu il est possible d'installer des fractions de coupleurs correspondant exactement au besoin. Par exemple si dans une solution on a $\mathbf{z}_i^1 = 1$, il suffit d'avoir $\mathbf{w}_i^1 = 1/cible_1$ pour satisfaire la contrainte (6.15).

Il serait donc intéressant d'étendre les inégalités valides de la section précédente de façon à inclure les variables \mathbf{w}_i^1 . En utilisant les nouvelles contraintes de dimensionnement (6.15) et (6.16) avec les inégalités (6.9) et (6.11) donnent les inégalités suivantes :

$$\sum_{i \in A} a_i \leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 + \frac{c^2}{\text{cible}_2} \sum_{i \in A \cap S} \mathbf{w}_i^2 \quad (6.17)$$

$$\sum_{i \in A} a_i \leq \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 + c^2 \mathbf{f}_{ji}^2) + \frac{c^2 c^1}{\text{cible}_1} \sum_{i \in A \cap S} \mathbf{w}_i^1 \quad (6.18)$$

En appliquant alors le principe détaillé dans la section (6.2.2.1), on obtient les inégalités valides suivantes :

$$\left(a - \left\lfloor \frac{\text{cible}_2 a}{c^2} \right\rfloor \frac{c^2}{\text{cible}_2} \right) \left\lceil \frac{\text{cible}_2 a}{c^2} \right\rceil \leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^3 + \left(a - \left\lfloor \frac{\text{cible}_2 a}{c^2} \right\rfloor \frac{c^2}{\text{cible}_2} \right) \sum_{i \in A \cap S} \mathbf{w}_i^2 \quad (6.19)$$

$$\left(a - \left\lfloor \frac{\text{cible}_1 a}{c^2 c^1} \right\rfloor \frac{c^2 c^1}{\text{cible}_1} \right) \left\lceil \frac{\text{cible}_1 a}{c^2 c^1} \right\rceil \leq \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 + c^2 \mathbf{f}_{ji}^2) + \left(a - \left\lfloor \frac{\text{cible}_1 a}{c^2 c^1} \right\rfloor \frac{c^2 c^1}{\text{cible}_1} \right) \sum_{i \in A \cap S} \mathbf{w}_i^1 \quad (6.20)$$

Dans l'exemple de la figure 6.6, on pourrait rajouter l'inégalité (6.18) pour l'ensemble $A = \{1, 2\}$:

$$\begin{aligned} & \left(112 - \left\lfloor \frac{0,16 \times 112}{8 \times 8} \right\rfloor \frac{8 \times 8}{0,16} \right) \left\lceil \frac{0,16 \times 112}{8 \times 8} \right\rceil \leq \mathbf{f}_{32}^3 + 8\mathbf{f}_{32}^2 + \left(112 - \left\lfloor \frac{0,16 \times 112}{8 \times 8} \right\rfloor \frac{8 \times 8}{0,16} \right) (\mathbf{w}_1^1 + \mathbf{w}_2^1) \\ \Leftrightarrow & 112 \leq \mathbf{f}_{32}^3 + 8\mathbf{f}_{32}^2 + 112 \times (\mathbf{w}_1^1 + \mathbf{w}_2^1) \end{aligned}$$

Dans l'exemple, le terme de droite vaut $112 \times 0,32 < 112$. L'inégalité rajoutée coupe donc cette solution fractionnaire.

Des comparaisons avec les solutions du modèle avec dimensionnement à 100% permettront de voir les économies réalisées dans l'achat de coupleurs, et de juger de la validité des règles actuelles de localisation des coupleurs de niveaux 1 et 2. Le taux de pénétration à prendre en compte pour chaque niveau de coupleurs fait partie des paramètres à passer au modèle.

On cherche en fait ici à dimensionner de façon prudente en tenant compte de l'incertitude de la demande finale. Plus on sous-estime la demande, plus les investissements seront chers sur le long terme mais moins on investit en premier lieu. La figure 6.7 illustre ce fait.

Pour un taux de pénétration final donné, il est possible d'établir une courbe présentant le coût de déploiement initial et le coût final selon le taux de pénétration initial pris en compte. Avec une meilleure connaissance de la distribution des clients et de son évolution, il pourrait être intéressant d'optimiser le dimensionnement des coupleurs de façon robuste sur plusieurs périodes.

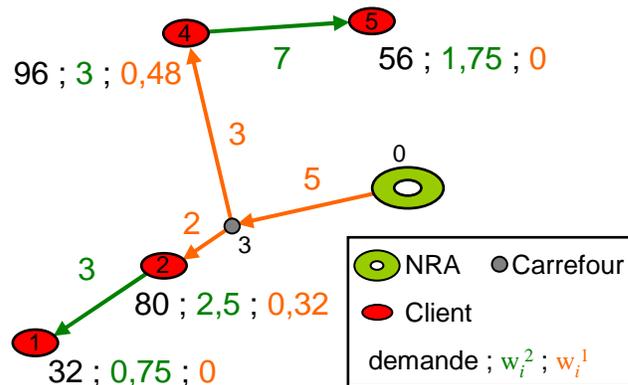


FIGURE 6.6 – Exemple de solution de la relaxation continue de (\mathcal{LCOdim}) avec des variables fractionnaires, avec des coupleurs 1 :8 et des taux de dimensionnement de 16% et 25%

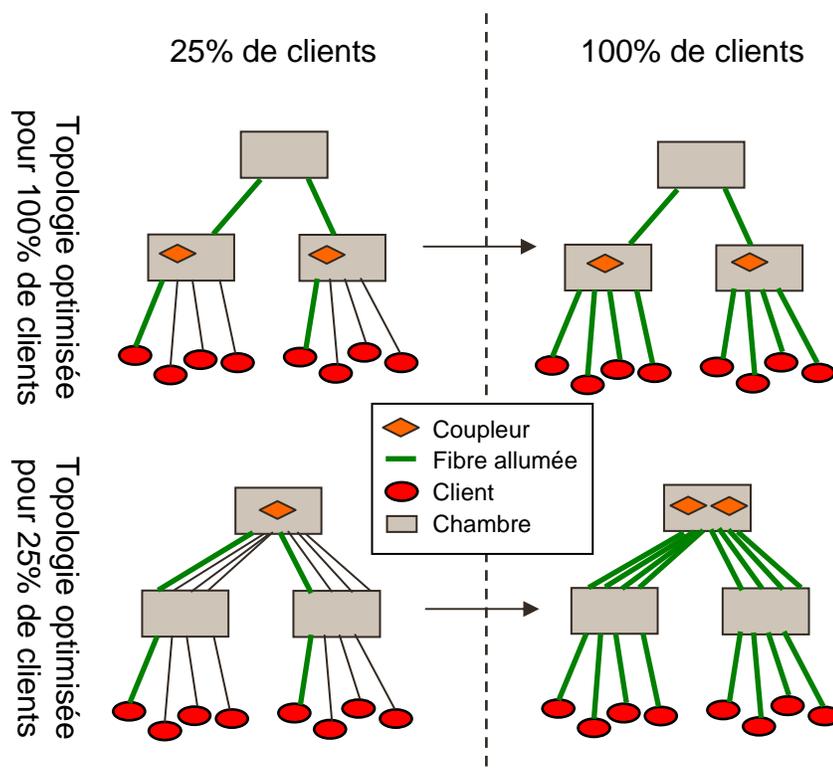


FIGURE 6.7 – Comparaison des déploiements selon le nombre de clients

6.2.5 Dimensionnement de l'OLT

Les différents composants installés au niveau de l'équipement central dépendent, entre autres, du nombre de fibres de niveau 1 utilisées en sortie de l'OLT, c'est à dire du nombre de coupleurs de niveau 1 effectivement installés. Chaque coupleur de niveau 1 installé doit être alimenté par une fibre de niveau 1 rattachée à une carte PON installée au niveau de l'OLT. Le nombre de fibres pouvant être reliées à chaque carte PON est limité.

Les coupleurs de niveau 1 ont donc un coût caché lié à l'installation des cartes PON. Il est possible de prendre en compte ce coût de différentes façons : soit en rajoutant une variable de décision représentant le nombre de cartes pon installées, soit en intégrant le coût d'une fraction de carte pon directement dans le coût d'installation d'un coupleur de niveau 1. En l'absence de coupes efficaces sur la nouvelle variable, ces deux méthodes reviennent au même car au continu la variable prendra systématiquement la valeur fractionnaire correspondant exactement au nombre de coupleurs de niveau 1 installés. De plus, la prise en compte de ce coût dégrade un peu plus encore la qualité de la borne obtenue par relaxation continue.

La prise en compte de ces coûts dans notre modèle ne peut donc s'envisager que si nous disposons de méthodes de résolution très efficaces.

6.2.6 Architecture à trois niveaux de coupleurs

Nous avons dit dans la section 6.1.1 que nous souhaitions un taux de partage de 1 : 64 (soit 64 clients au maximum par fibre connectée à une carte PON). Ce taux de partage est atteint dans les pilotes avec deux niveaux de coupleurs. Mais une autre architecture avec trois niveau de coupleurs est également possible. Le premier niveau de coupleurs serait placé directement au niveau de l'OLT, comme illustré sur la figure 6.8.

La localisation des coupleurs de niveau 0 étant déjà fixée, il est possible de résoudre le problème de localisation de coupleurs à trois niveaux de façon approchée en utilisant le modèle de base à deux niveaux pour localiser les coupleurs des niveaux 1 et 2. Ensuite, le nombre de coupleurs de niveau 0 nécessaires est calculé en fonction du nombre de coupleurs de niveau 1 installés. Le coût des coupleurs de niveau 0 étant proche de ceux des autres niveaux tout en étant beaucoup moins nombreux, ils ne représentent qu'une faible fraction du coût total. Il ne devraient donc avoir qu'un faible impact sur la topologie optimale.

6.2.7 Création de nouveau génie civil et connexions point-à-point

Dans des zones où il y a peu de génie civil existant, il n'y a pas d'autre choix que de poser de nouvelles conduites. La création de nouvelles tranchées doit donc être

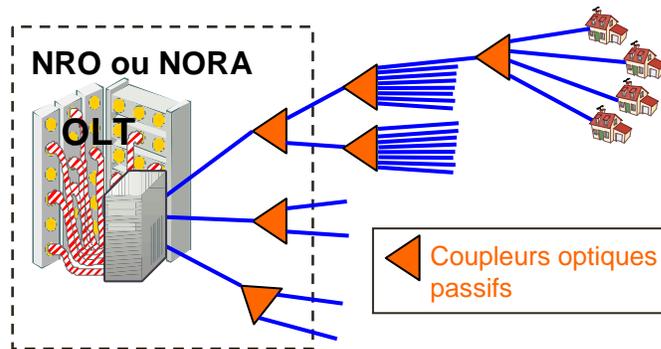


FIGURE 6.8 – Architecture GPON à trois niveaux de coupleurs optiques

intégrée au problème. Les sections suivantes présentent une variante de modélisation prenant les tranchées en compte.

Nous en profitons également pour proposer dans le modèle qui suit la possibilité de raccorder certains clients à l'OLT sans passer nécessairement par 2 niveaux de coupleurs.

En plus de celles du modèle de base (voir section 6.2.2), on définit les variables et les notations suivantes :

- \mathbf{f}_i^k $k = 1, 2, 3$: variables représentant le nombre de fibres de niveau k « consommées » par le nœud i (fibres utiles pour satisfaire la demande, ou fibres non utilisées en sortie d'un coupleur) ;
- \mathbf{y}_{ij} : variable binaire indiquant le creusement d'une tranchée sur l'arête $(i, j) \in E$;
- T : le coût au mètre d'une tranchée ;
- M : nombre maximum de fibres pouvant être mises dans une tranchée (grande constante simulant l'infini) ;

Le problème de localisation de coupleurs optiques peut alors se modéliser sous la forme du programme linéaire en nombres entiers suivant :

$$\begin{array}{l}
\min_{\mathbf{F}, \mathbf{Y}, \mathbf{z}} \quad \sum_{i \in \mathcal{S}} (C^1 \mathbf{z}_i^1 + C^2 \mathbf{z}_i^2) + \sum_{(i,j) \in E} d_{ij} \left(T \mathbf{y}_{ij} + \sum_{k=1}^3 F^k (\mathbf{f}_{ij}^k + \mathbf{f}_{ji}^k) \right) \\
\text{s.c. :} \\
\mathbf{z}_i^1 + \mathbf{f}_i^1 = \sum_{j \neq i} \mathbf{f}_{ji}^1 - \sum_{j \neq i} \mathbf{f}_{ij}^1 \quad \forall i \in \mathcal{S} \setminus \{0\} \quad (6.21) \\
\mathbf{z}_i^2 + \mathbf{f}_i^2 = \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 + c^1 \mathbf{z}_i^1 \quad \forall i \in \mathcal{S} \quad (6.22) \\
\mathbf{f}_i^3 = \sum_{j \neq i} \mathbf{f}_{ji}^3 - \sum_{j \neq i} \mathbf{f}_{ij}^3 + c^2 \mathbf{z}_i^2 \quad \forall i \in \mathcal{S} \quad (6.23) \\
\mathbf{f}_i^1 = \sum_{j \neq i} \mathbf{f}_{ji}^1 - \sum_{j \neq i} \mathbf{f}_{ij}^1 \quad \forall i \in V \setminus (\mathcal{S} \cup \{0\}) \quad (6.24) \\
\mathbf{f}_i^2 = \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 \quad \forall i \in V \setminus \mathcal{S} \quad (6.25) \\
\mathbf{f}_i^3 = \sum_{j \neq i} \mathbf{f}_{ji}^3 - \sum_{j \neq i} \mathbf{f}_{ij}^3 \quad \forall i \in V \setminus \mathcal{S} \quad (6.26) \\
\mathbf{f}_i^1 + \mathbf{f}_i^2 + \mathbf{f}_i^3 \geq a_i \quad \forall i \in \mathcal{C} \quad (6.27) \\
\sum_{k=1}^3 (\mathbf{f}_{ij}^k + \mathbf{f}_{ji}^k) \leq b_{ij} + M \mathbf{y}_{ij} \quad \forall (i, j) \in E \quad (6.28) \\
\mathbf{f}_{ij}^k, \mathbf{f}_i^k, \mathbf{z}_i^k \in \mathbb{N}; \mathbf{y}_{ij} \in \{0, 1\}
\end{array}$$

Avec l'ajout des variables \mathbf{f}_i^k , les contraintes de flots se transforment toutes en égalités. Les contraintes (6.21) et (6.24) assurent :

- que chaque coupleur de niveau 1 est alimenté par une fibre de niveau 1 ;
- la conservation du flot de niveau 1 lorsque le sommet i n'est pas un client et qu'aucun coupleur n'y est installé ;
- la possibilité de satisfaire la demande d'un client situé en i grâce à des fibres de niveau 1 (*i.e.* lorsque que \mathbf{f}_i^1 est non nulle).

Les contraintes des niveaux 2 et 3 suivent un principe similaire.

La satisfaction de la demande est maintenant assurée par les contraintes (6.27) où la demande peut être satisfaite par des fibres optiques de l'un ou l'autre niveau. Les contraintes de capacités changent également et font maintenant intervenir les variables \mathbf{y}_{ij} . Ainsi, si une tranchée a été creusée sur l'arête (i, j) sa capacité passe à $b_{ij} + M$. Cette modification dégrade fortement la qualité de la borne obtenue par relaxation continue, d'autant plus que le coût des tranchées est souvent bien plus important que celui des fibres ou des coupleurs.

Cette constatation nous amène soit à chercher des inégalités valides pour les variables

\mathbf{y}_{ij} , soit à élaborer des heuristiques fixant d'abord les tranchées avant de repasser au modèle de base.

6.2.7.1 Inégalités valides sur les flots

Le fait de pouvoir satisfaire la demande avec n'importe quel niveau de fibres rend moins évidente la recherche d'inégalité valides d'après le principe détaillé dans la section (6.2.2.1). Il faut en effet additionner les trois variables \mathbf{f}_i^1 , \mathbf{f}_i^2 , et \mathbf{f}_i^3 pour pouvoir faire apparaître la demande du sommet i .

1. Additionnons l'ensemble des contraintes de flots pour $i \in A$ et notons : $\mathbf{f}_i = \mathbf{f}_i^1 + \mathbf{f}_i^2 + \mathbf{f}_i^3$, $\mathbf{f}_{ij} = \mathbf{f}_{ij}^1 + \mathbf{f}_{ij}^2 + \mathbf{f}_{ij}^3$.

On obtient :

$$\sum_{i \in A} \mathbf{f}_i + \sum_{i \in A, j \notin A} \mathbf{f}_{ij} = \sum_{i \in A, j \notin A} \mathbf{f}_{ji} + \sum_{i \in A} (c^1 - 1) \mathbf{z}_i^1 + \sum_{i \in A} (c^2 - 1) \mathbf{z}_i^2$$

En utilisant (6.27) et la positivité des flots, on obtient l'inégalité :

$$\sum_{i \in A} a_i \leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji} + \sum_{i \in A} (c^1 - 1) \mathbf{z}_i^1 + \sum_{i \in A} (c^2 - 1) \mathbf{z}_i^2 \quad (6.29)$$

Dans le cas où $c^1 = c^2 = c$, en posant $\mathbf{x} = \sum_{i \in A} (\mathbf{z}_i^1 + \mathbf{z}_i^2)$, $\mathbf{y} = \sum_{i \in A, j \notin A} \mathbf{f}_{ji}$, et $a = \sum_{i \in A} a_i$ on obtient une inégalité de la forme $(c - 1)\mathbf{x} + \mathbf{y} \geq a$ avec $\mathbf{x}, \mathbf{y} \in \mathbb{N}$. On peut alors créer une inégalité valide selon la formule de la section (6.2.2.1).

2. Additionnons les contraintes de flot de niveau 2 pour $i \in A$. On obtient :

$$\sum_{i \in A} \mathbf{z}_i^2 + \sum_{i \in A} \mathbf{f}_i^2 + \sum_{i \in A, j \notin A} \mathbf{f}_{ij}^2 = \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^2 + c^1 \sum_{i \in A} \mathbf{z}_i^1$$

Additionnons $\sum_{i \in A} \mathbf{f}_i^1 + \mathbf{f}_i^3$ de part et d'autre. On obtient :

$$\sum_{i \in A} \mathbf{z}_i^2 + \sum_{i \in A} (\mathbf{f}_i^1 + \mathbf{f}_i^2 + \mathbf{f}_i^3) + \sum_{i \in A, j \notin A} \mathbf{f}_{ij}^2 = \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^2 + c^1 \sum_{i \in A} \mathbf{z}_i^1 + \sum_{i \in A} (\mathbf{f}_i^1 + \mathbf{f}_i^3)$$

En utilisant (6.27), la positivité des flots et des variables \mathbf{z}_i^2 , on obtient l'inégalité :

$$\sum_{i \in A} a_i \leq \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^2 + c^1 \sum_{i \in A} \mathbf{z}_i^1 + \sum_{i \in A} (\mathbf{f}_i^1 + \mathbf{f}_i^3) \quad (6.30)$$

En posant $\mathbf{x} = \sum_{i \in A} \mathbf{z}_i^1$, $\mathbf{y} = \sum_{i \in A, j \notin A} \mathbf{f}_{ji}^2 + \sum_{i \in A} (\mathbf{f}_i^1 + \mathbf{f}_i^3)$, $a = \sum_{i \in A} a_i$ on obtient une inégalité de la forme $c^1 \mathbf{x} + \mathbf{y} \geq a$ avec $\mathbf{x}, \mathbf{y} \in \mathbb{N}$. On peut alors créer une inégalité valide selon la formule du paragraphe précédent.

3. On pourrait agir de façon similaire avec les contraintes de flots de niveau 3.

6.2.7.2 Inégalités valides basées sur les capacités

Pour renforcer la formulation, il est possible d'ajouter des inégalités inspirées des inégalités de coupes pour le problème de l'arbre de Steiner. Soit A un ensemble de sommets $i \in V \setminus \{0\}$. Posons $b = \sum_{i \in A, j \notin A, (i,j) \in E} b_{ij}$ et $a = \sum_{i \in A} a_i$. b est la somme des capacités incidentes à l'ensemble A et a est la somme des demandes des sommets de A .

Propriété 6.5. *Si $b < \frac{a}{c^1 c^2}$ alors il faut nécessairement construire une tranchée incidente à A ce qui s'exprime par l'inégalité valide suivante :*

$$\sum_{i \in A, j \notin A, (i,j) \in E} \mathbf{y}_{ij} \geq 1 \quad (6.31)$$

Démonstration

Sommons chaque niveau de contraintes de flots pour l'ensemble des sommets de A . Nous obtenons les trois égalités suivantes :

$$\sum_{i \in A} \mathbf{f}_i^3 = \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 - \mathbf{f}_{ij}^3) + c^2 \sum_{i \in A} \mathbf{z}_i^2 \quad (6.32)$$

$$\sum_{i \in A} \mathbf{z}_i^2 = \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^2 - \mathbf{f}_{ij}^2) + c^1 \sum_{i \in A} \mathbf{z}_i^1 - \sum_{i \in A} \mathbf{f}_i^2 \quad (6.33)$$

$$\sum_{i \in A} \mathbf{z}_i^1 = \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^1 - \mathbf{f}_{ij}^1) - \sum_{i \in A} \mathbf{f}_i^1 \quad (6.34)$$

En remplaçant \mathbf{z}_i^1 dans (6.33), puis \mathbf{z}_i^2 dans (6.32), on obtient :

$$\begin{aligned} \sum_{i \in A} \mathbf{f}_i^3 &= \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^3 - \mathbf{f}_{ij}^3) + c^2 \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^2 - \mathbf{f}_{ij}^2) + c^1 c^2 \sum_{i \in A, j \notin A} (\mathbf{f}_{ji}^1 - \mathbf{f}_{ij}^1) - c^2 \sum_{i \in A} \mathbf{f}_i^2 - c^1 \sum_{i \in A} \mathbf{f}_i^1 \\ &\Rightarrow \sum_{i \in A} (\mathbf{f}_i^3 + c^2 \mathbf{f}_i^2 + c^1 c^2 \mathbf{f}_i^1) \leq c^1 c^2 \sum_{i \in A, j \notin A} \sum_{k=1}^3 (\mathbf{f}_{ji}^k - \mathbf{f}_{ij}^k) \end{aligned}$$

$$\text{avec (6.27)} \Rightarrow \sum_{i \in A} a_i \leq c^1 c^2 \sum_{i \in A, j \notin A} \sum_{k=1}^3 \mathbf{f}_{ji}^k$$

$$\Rightarrow \sum_{i \in A} a_i \leq c^1 c^2 \left(M \sum_{i \in A, j \notin A} \mathbf{y}_{ij} + \sum_{i \in A, j \notin A} b_{ij} \right)$$

Si on a $b < \frac{a}{c^1 c^2}$, alors :

$$\begin{aligned} \sum_{i \in A} a_i &< c^1 c^2 \left(M \sum_{i \in A, j \notin A} \mathbf{y}_{ij} + \sum_{i \in A, j \notin A} b_{ij} \right) \\ \Rightarrow \sum_{i \in A, j \notin A} \mathbf{y}_{ij} &\geq \frac{\sum_{i \in A} a_i / c^1 c^2 - \sum_{i \in A, j \notin A} b_{ij}}{M} > 0 \end{aligned}$$

Et donc :

$$b < \frac{a}{c^1 c^2} \Rightarrow \sum_{i \in A, j \notin A} \mathbf{y}_{ij} \geq 1 \quad (6.35)$$

□

Soit A_1 et A_2 deux ensembles de sommets tels que $\{(i, j) \mid i \in A_1, j \in A_2\} = \emptyset$, il est facile de montrer que les inégalités valides (6.31) pour A_1 et A_2 impliquent celle pour $A_1 \cup A_2$. Ainsi, les inégalités (6.31) ne sont intéressantes à générer que pour des ensembles de sommets induisant un sous-graphe connexe.

Si le graphe sous-jacent au problème comporte des arêtes sans capacité initiale, il est possible d'améliorer les inégalités (6.31).

Propriété 6.6. Si $\sum_{i \in A, j \notin A, (i, j) \in E} b_{ij} < \frac{a}{64}$ et $\sum_{i \in A, j \in A, (i, j) \in E} b_{ij} = 0$, et si $a_i > 0$ pour tout $i \in A$, l'inégalité suivante est valide pour le problème (\mathcal{LCO}_{gc}) :

$$\sum_{i \in A, j \notin A, (i, j) \in E} \mathbf{y}_{ij} + \sum_{i \in A, j \in A, (i, j) \in E} \mathbf{y}_{ij} \geq |A| \quad (6.36)$$

Démonstration : on procède par récurrence sur le nombre d'éléments de A .

$|A| = 1$. Supposons $A = \{i\}$, l'inégalité (6.36) devient $\sum_{j \notin A, (i, j) \in E} \mathbf{y}_{ij} \geq 1$ ce qui correspond à l'inégalité (6.31) présentée plus haut.

$|A| \geq 2$. Pour $k \in A$ notons $A_k = A \setminus \{k\}$. Pour tout $k \in A$, par hypothèse de récurrence, l'inégalité suivante

$$\sum_{i \in A_k, j \notin A_k, (i, j) \in E} \mathbf{y}_{ij} + \sum_{i \in A_k, j \in A_k, (i, j) \in E} \mathbf{y}_{ij} \geq |A_k| = |A| - 1$$

est valide. Additionnons ces $|A|$ inégalités avec $\sum_{i \in A, j \notin A, (i, j) \in E} \mathbf{y}_{ij} \geq 1$ l'inégalité (6.31) présentée plus haut. Dans cette somme, chaque arête (i, j) avec $i \in A, j \notin A$ ou $i \in A, j \in A$ apparaît $|A|$ fois. On obtient donc

$$\begin{aligned} |A| \times \left(\sum_{i \in A, j \notin A, (i, j) \in E} \mathbf{y}_{ij} + \sum_{i \in A, j \in A, (i, j) \in E} \mathbf{y}_{ij} \right) &\geq |A| \times (|A| - 1) + 1 \\ \Leftrightarrow \sum_{i \in A, j \notin A, (i, j) \in E} \mathbf{y}_{ij} + \sum_{i \in A, j \in A, (i, j) \in E} \mathbf{y}_{ij} &\geq |A| - 1 + \frac{1}{|A|} \end{aligned}$$

Le membre de gauche étant un nombre entier, on peut arrondir le membre de droite $|A| - 1 + \frac{1}{|A|}$ au premier entier supérieur, soit $|A|$.

6.2.7.3 Heuristique à base d'arbres

Dans les zones peu denses où le génie civil existant est moins abondant, il est probable que beaucoup de nouvelles tranchées soient nécessaires. Étant donné que le coût d'un mètre de tranchée est bien supérieur à celui d'un mètre de fibre et au prix d'un coupleur, il est probable que le génie civil forme la plus grosse part du coût total.

Cette remarque nous amène à considérer des heuristiques décomposant le problème en deux : déterminer les nouvelles tranchées, puis intégrer ces tranchées aux capacités de chaque arc et repasser au modèle de base. Le premier problème consistera à se donner un arbre reliant tous les clients à l'OLT par des tranchées. Les variables y_{ij} sont alors fixées et le modèle de base obtenu (sans génie civil). Enfin, si des tranchées de l'arbre ne sont pas utilisées, on peut les retirer de la solution finale.

Plusieurs possibilités existent pour le choix de l'arbre : arbre des plus courts chemins entre les clients et l'OLT ; arbre couvrant sur tout le graphe ; arbre de Steiner dont les terminaux sont les clients et l'OLT ; etc...

À l'heure actuelle, nous n'avons implémenté que la variante avec l'arbre couvrant. Celle-ci permet de traiter des instances à l'échelle d'une zone locale en quelques minutes.

6.2.8 Expérimentations numériques

Nous avons testés nos différents modèles sur de petites instances fictives mais également sur des instances issues de données réelles. Les tests ont été réalisés avec le solveur CPLEX 11 sur un serveur linux équipé d'un processeur xeon 3,6Ghz et de 4Go de mémoire vive.

6.2.8.1 Modèle sans nouveau génie civil

Le modèle de base sans création de nouveau génie civil a été testé sur deux instances issues de données réelles, nommées GLO et BDV, correspondant à deux zones urbaines respectivement dense et très dense.

Le tableau (6.1) présente les caractéristiques des deux instances. On constate que le nombre d'arêtes et de sommets du graphe de GLO sont plus grands que pour BDV, mais que la demande totale et la demande moyenne par sommet y sont inférieures. Cela correspond bien à la différence de densité de population entre ces deux instances.

Nous avons testé ces deux instances avec et sans prise en compte du dimensionnement des coupleurs. Dans le cas avec dimensionnement des coupleurs, en plus de

Instance	GLO	BDV
Nb. de sommets ¹	1623	807
Nb. d'arêtes ¹	1987	1101
Demande totale	23774	49294
Demande moyenne	14.6412	57.3656

TABLE 6.1 – Présentation des principales caractéristiques des instances GLO et BDV

la résolution standard par CPLEX, nous avons implémenté un branch&cut avec les inégalités valides (6.20) seules ou avec (6.19). Dans tous les cas, nous avons fixé une limite de temps d'une heure pour la résolution.

Pour observer l'utilisation des coupleurs de niveau 1 et 2, nous avons défini deux indicateurs nommé taux d'utilisation client et taux d'utilisation de la capacité. Le taux d'utilisation client pour un coupleur est défini comme le rapport entre le nombre de clients raccordés au coupleur et le nombre maximum de clients pouvant être raccordé au coupleur. Le taux d'utilisation de la capacité est quant à lui défini par le rapport entre le nombre de sorties utilisées sur le coupleur et la capacité de ce dernier. Pour les coupleurs de niveau 2, les deux définitions reviennent aux même car chaque sortie utilisée correspond à une fibre de niveau 3 qui sera raccordée à un client. En revanche, pour les coupleurs de niveau 1, les deux définitions diffèrent. En effet, le lien entre les sorties utilisées et le nombre de clients raccordés dépend du taux d'utilisation client des coupleurs de niveau 2.

Il n'est pas possible avec notre modèle de base, de savoir exactement quels coupleurs de niveau 2 sont reliés à quels coupleurs de niveau 1. Pour les coupleurs de niveau 1, nous nous sommes donc contentés de calculer le taux moyen d'utilisation de la capacité pour chaque sommet i à l'aide des formules suivantes :

$$\text{Sans dimensionnement : } \frac{\sum_{j \neq i} \mathbf{f}_{ij}^2 - \sum_{j \neq i} \mathbf{f}_{ji}^2 + \mathbf{z}_i^2}{c^1 \mathbf{z}_i^1} \quad (6.37)$$

$$\text{Avec dimensionnement : } \frac{\text{cible}^1 \left(\sum_{j \neq i} \mathbf{f}_{ij}^2 - \sum_{j \neq i} \mathbf{f}_{ji}^2 + \mathbf{z}_i^2 \right)}{c^1 \mathbf{w}_i^1} \quad (6.38)$$

Pour les coupleurs de niveau 2, nous pouvons calculer le taux d'utilisation client moyen pour le sommet i avec les formules suivantes :

$$\text{Sans dimensionnement : } \frac{\sum_{j \neq i} \mathbf{f}_{ij}^3 - \sum_{j \neq i} \mathbf{f}_{ji}^3 + a_i}{c^2 \mathbf{z}_i^2} \quad (6.39)$$

$$\text{Avec dimensionnement : } \frac{\text{cible}^2 \left(\sum_{j \neq i} \mathbf{f}_{ij}^3 - \sum_{j \neq i} \mathbf{f}_{ji}^3 + a_i \right)}{c^2 \mathbf{w}_i^2} \quad (6.40)$$

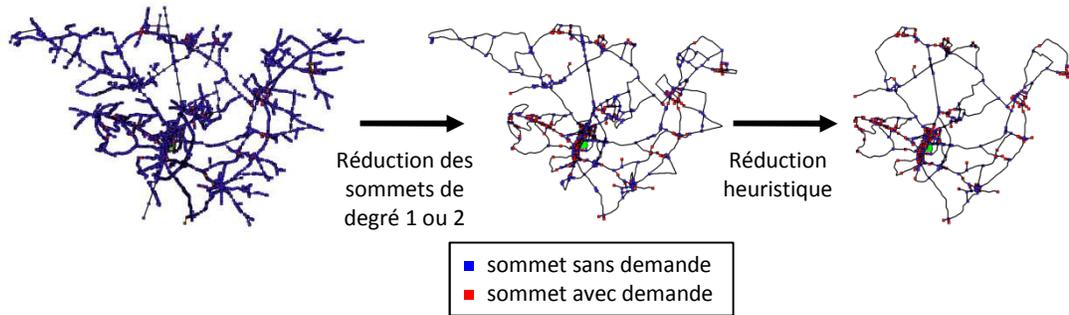


FIGURE 6.9 – Effet des différentes réductions du graphe sur l’instance AIG

Ces deux critères fournissent un autre moyen d’apprécier et de comprendre la qualité des solutions trouvées.

Dans les tests, les capacités des coupleurs de niveau 1 et 2 sont identiques et égales à 8. Les coûts des coupleurs et des fibres optiques sont les coûts réels.

Sans dimensionnement des coupleurs Lors de la résolution du modèle sans dimensionnement, CPLEX n’a trouvé de solution optimale pour aucune des deux instances. Mais la qualité des solutions trouvées est assez bonne. Nous n’avons donc pas testé les coupes sur ce modèle et nous sommes passés directement aux tests avec dimensionnement.

Les résultats des tests

Avec dimensionnement des coupleurs Dans les tests avec dimensionnement, le taux de pénétration cible de niveau 2 a été fixé à 25%, et celui de niveau 1 à 16%.

6.2.8.2 Modèle avec création de génie civil

Le modèle avec création de génie civil a été testé sur une instance réelle pour laquelle les données sur le génie civil existant ne sont pas disponibles. C’est donc le graphe de la voirie qui est utilisé. L’instance AIG correspond à une zone rurale. Comme les données sont manquantes, les capacités existantes sont considérées comme nulles.

Sur la figure 6.9, on peut constater l’efficacité des réductions du graphe dans le cas de l’instance AIG.

L’heuristique à base d’arbre couvrant a été utilisée sur AIG et la topologie de la solution obtenue est illustrée dans la figure 6.10. Cette heuristique permet de trouver des solutions rapidement pour des instances de taille réelle. Malheureusement, la qualité de la borne de la relaxation continue pour le problème avec création de génie

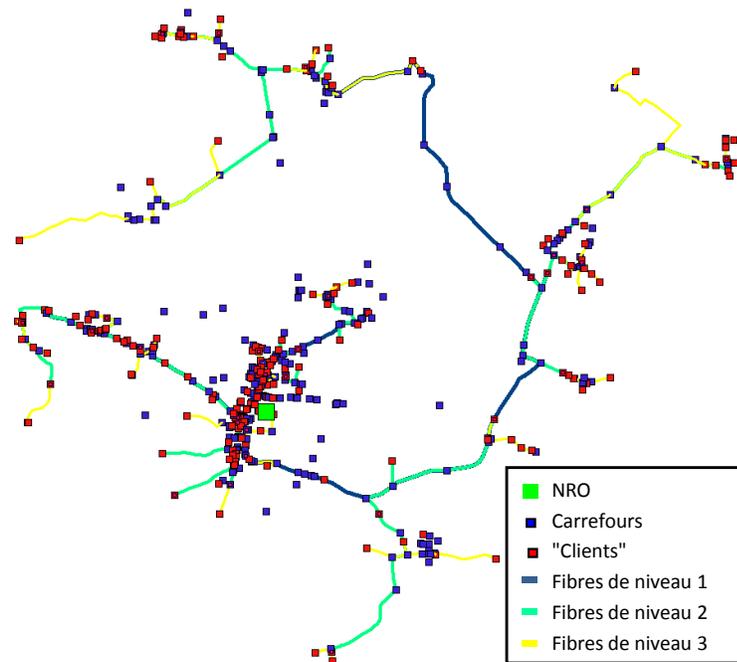


FIGURE 6.10 – Topologie de la solution heuristique de déploiement pour AIG

civil est très mauvaise, et nous n'avons pas implémenté de branch-&-cut avec les coupes présentées dans la section 6.2.7.2. Nous ignorons donc la qualité des solutions de l'heuristique.

6.2.9 Perspectives

Comme le montre les résultats numériques sur les deux instances réelles, des progrès restent à faire sur les méthodes de résolution pour obtenir de bonnes solutions sur tout type d'instance. Pour cela, il serait intéressant d'élaborer de bonnes heuristiques pouvant éventuellement être mises à profit dans l'algorithme de branch&cut.

Si les méthodes de résolution progressent, il serait alors intéressant de chercher à intégrer dans les modèles la contrainte de distance maximum entre un abonné et l'OLT. Il n'est pas possible de le faire avec le modèle de base actuel car celui-ci ne permet pas d'établir le routage exact du trafic entre un abonné et l'OLT. En effet, si des fibres arrivent à un noeud par plusieurs endroits, le modèle actuel ne fait pas de différence entre la provenance des fibres sortant de ce noeud. Une formulation à l'aide d'un multi-flot permettrait de remédier à cela, mais les coupleurs optiques compliquent la tâche. De plus, il faudrait un flot par PRI, ce qui peut provoquer une explosion du nombre de variables sur les problèmes de tailles réelles. Heureusement

les graphes de nos instances sont peu denses, mais est ce suffisant ?

Un avantage à obtenir un routage précis des trafics dans une solution du problème de localisation de coupleurs serait la simplification du passage au problème de câblage. Ces améliorations sont nécessaires car les solutions actuelles ne fournissent pas assez de renseignements pour étudier en profondeur les règles d'ingénierie. Un indicateur très utile serait la distance moyenne entre un client et son coupleur de niveau 2. Pour l'instant nous sommes capables d'obtenir avec la solution de (*LCO*) le pourcentage de clients pour lesquels cette distance est nulle. C'est une première indication mais ce n'est pas suffisant.

6.3 Problème d'affectation du câblage pour l'accès FTTH

Un certain nombre d'hypothèses simplificatrices sont réalisées dans le problème de localisation de coupleurs optiques et certaines décisions concernant le déploiement du réseau sont laissées en suspens. C'est le cas notamment du choix des câbles. En effet, la solution du premier problème ne prend pas en compte le fait que les fibres optiques ne sont pas installées une à une mais par câbles de capacités variables.

Le choix des câbles a un impact économique mais influe également sur la qualité du signal, donc sur la portée maximale, et donc sur l'éligibilité ou non d'un abonné à une offre. Le choix d'un câblage est donc un réel problème d'optimisation.

6.3.1 Contrainte d'éligibilité

Un des avantages du réseau d'accès FTTH est qu'il permet une plus grande portée que l'actuel réseau cuivre. Cela devrait permettre d'accroître le taux d'éligibilité des clients sur un territoire. Certains facteurs atténuent le signal entre l'abonné et l'équipement central et font diminuer cette portée. On dit qu'ils consomment du budget optique. Le budget optique représente l'atténuation maximale du signal tolérée par l'architecture entre un client et l'OLT. Ce budget est consommé par le passage dans les coupleurs, les épissures (raccords) entre les fibres optiques, les différents connecteurs, est la distance entre les clients et l'OLT.

Le passage par un coupleur optique passif atténue grandement le signal. Mais comme le signal destiné à chaque abonné passe par un nombre fixé de coupleurs, cette atténuation ne dépend pas du câblage. En revanche, les soudures qui sont réalisées entre les fibres de deux câbles différents qui sont raccordés l'un à l'autre dépendent directement du choix du câblage.

Ce problème d'optimisation du câblage est donc soumis à une contrainte sur le budget optique qui limite le nombre d'épissures entre les clients et l'OLT, afin de respecter les critères d'éligibilité pour chaque client.

6.3.2 Arborescence de câblage

Pour pouvoir traiter ce problème, la solution de déploiement est représentée par une arborescence. La racine de cette arborescence est le NRO. À partir du NRO, une arête est créée à chaque fois que des fibres d'un niveau vont vers un sommet.

Cette arborescence représente l'ensemble des routages possibles pour le trafic entre chaque client et l'OLT. Certains sommets peuvent donc être représentés plusieurs fois s'ils sont accessibles de différentes façons depuis l'équipement central. On peut observer ce phénomène sur l'exemple des figures 6.11, 6.12, et 6.13 avec le sommet 11.

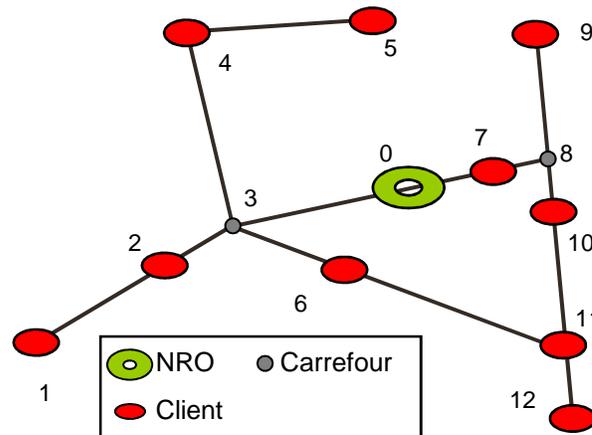


FIGURE 6.11 – Exemple de topologie du génie civil existant

La description de l'instance du problème de câblage avec cette arborescence doit nous permettre de connaître pour tout sommet les deux facteurs qui jouent sur le budget optique et qui ne dépendent pas du câblage :

La portée. La longueur de l'unique chemin entre l'équipement central et un noeud donné est connue.

Les coupleurs. Pour chaque noeud, on connaît le nombre de coupleurs par lequel on est passé.

Cela implique que lorsqu'un sommet peut être atteint par deux chemins différents, il faut créer deux sous-arborescences identiques. Dans la première on considère que les fibres sont passées par le premier chemin, et par le second chemin dans la seconde. De ce fait, des parties entières du sous-graphe utilisé par la solution de déploiement se trouvent dupliqués plusieurs fois dans l'arborescence.

Les étapes de construction de l'arborescence consistent à parcourir la solution de déploiement de la façon suivante :

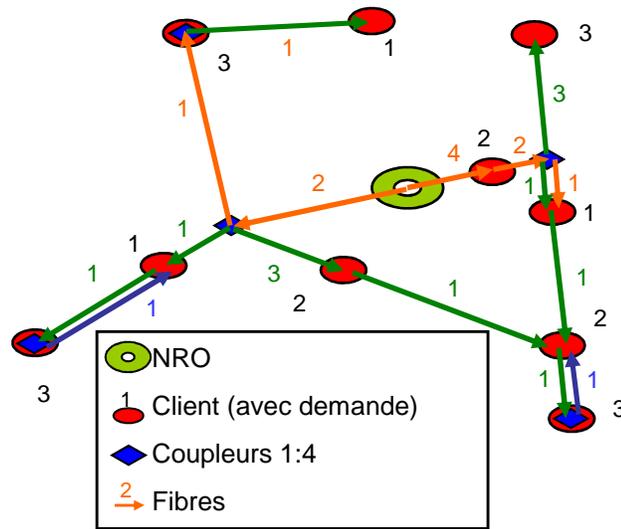


FIGURE 6.12 – Exemple de solution de déploiement

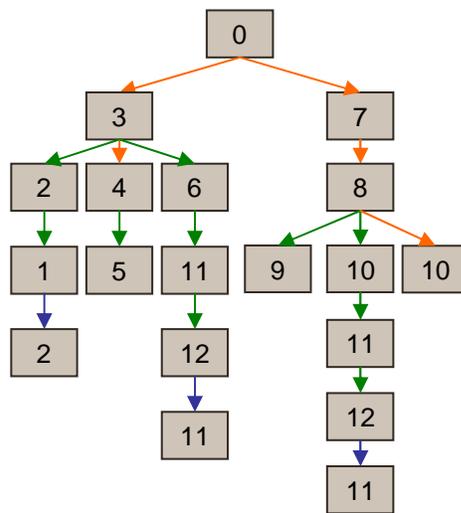


FIGURE 6.13 – Arborecence pour le déploiement de la figure 6.12

La liste des noeuds actifs est initialisée avec l'emplacement de l'OLT comme noeud de niveau 1 ;

while liste des noeuds actifs non vide **do**

Noeud courant \leftarrow Premier noeud de la liste ;

Niv \leftarrow niveau du noeud courant

On enlève le premier noeud de la liste ;

for all arêtes incidentes à *Noeud courant* **do**

if des fibres de niveau *Niv* passent sur l'arête **then**

Création de l'arc de niveau *Niv* correspondant dans l'arborescence ;

Rajout de l'extrémité de l'arête dans la liste des noeuds actif comme noeud de niveau *Niv* ;

end if

if $Niv \leq 2$ & des fibres de niveau $Niv + 1$ passent sur l'arête & un coupleur de niveau *Niv* est installé sur *Noeud courant* **then**

Création de l'arc de niveau $Niv + 1$ correspondant dans l'arborescence ;

Rajout de l'extrémité de l'arête dans la liste des noeuds actif comme noeud de niveau $Niv + 1$;

end if

if $Niv = 1$ & des fibres de niveau 3 passent sur l'arête & un coupleur de niveau 1 est installé sur *Noeud courant* & un coupleur de niveau 2 est installé sur *Noeud courant* **then**

Création de l'arc de niveau 3 correspondant dans l'arborescence ;

Rajout de l'extrémité de l'arête dans la liste des noeuds actif comme noeud de niveau 3 ;

end if

end for

end while

La propriété 6.4, qui dit qu'il n'y a aucun circuit de fibres de même niveau dans la solution optimale du problème de localisation de coupleurs optiques, assure la terminaison de l'algorithme ci-dessus. Toutefois, si l'on dispose seulement d'une solution sous-optimale dont les circuits n'ont pas été éliminés, il faut prendre garde dans l'algorithme à empêcher de retourner à un noeud de même niveau déjà exploré.

Propriété 6.7. *Lorsque les arêtes du graphe utilisées par le déploiement forment un arbre ayant n sommets, la construction de l'arborescence a une complexité en $O(n^3)$.*

Démonstration.

Nous allons calculer tout d'abord le nombre maximum d'arc de niveau 1 (*i.e.* d'arc sur lequel passent uniquement une ou des fibres de niveau 1). Dans un arbre, il n'y a qu'un seul chaîne reliant la racine à chaque sommet du graphe. D'après la propriété

6.3, il ne peut pas y avoir de fibres de même niveau circulant dans les deux sens sur cette chaîne. En conséquence, il y aura autant d'arcs de niveau 1 dans l'arborescence que d'arêtes dans l'arbre de la solution de déploiement, soit $n - 1$ arcs.

Calculons maintenant le nombre d'arc de niveau 2. Une sous-arborescence de niveau 2 peut avoir pour racine n'importe quelle extrémité d'un arc de niveau 1 ou la racine de l'arborescence. Pour chacune de ces sous-arborescences nous pouvons faire le même calcul que pour les arcs de niveau 1, prouvant qu'elles ont au maximum $n - 1$ arcs. Il y a donc au maximum $n(n - 1)$ arcs de niveau 2.

Avec les mêmes arguments que précédemment on montre qu'il y a au maximum $n \times n(n - 1)$ arcs de niveau trois.

L'algorithme de construction de l'arborescence créera donc au maximum $(n^2 + n + 1)(n - 1)$ arcs. \square

Cette complexité en $O(n^3)$ dans le cas d'un arbre peut paraître déjà très élevée. Mais il s'agit d'un pire cas. En pratique les arborescences de fibres de même niveau issues de l'OLT ou des coupleurs ne couvrent pas l'ensemble des arêtes du déploiement. En fait, peu d'arêtes sont utilisées en même temps par des fibres de niveaux différents. Par conséquent, dans le cas d'un arbre, on peut espérer une taille d'arborescence qui soit plus de l'ordre de $O(n)$.

Malheureusement, dans le cas général, rien ne garantit que la taille de l'arborescence soit une fonction polynomiale de la taille du sous-graphe utilisé par la solution de déploiement. Malgré cela, il faut rappeler que nos graphes sont très peu denses et que donc le sous-graphe utilisé pour le déploiement, s'il n'est pas un arbre, est très peu dense lui aussi.

Un exemple rassurant est celui de l'instance GLO utilisé dans la section précédente. À partir d'une bonne solution du problème de localisation de coupleurs sans dimensionnement de ces derniers nous avons pu construire une arborescence ayant environ 15000 arêtes, ce qui reste manipulable.

La principale raison de la grande taille de l'arborescence est le fait que le routage exact du trafic de chaque client n'est pas défini par la solution du problème de localisation de coupleurs. En effet, connaître ce routage nous permettrait d'éviter d'avoir à dupliquer certaines parties du sous-graphe dans l'arborescence.

6.3.2.1 Heuristique de réduction de l'arborescence

Il est possible de réduire l'arborescence de câblage en effectuant certains choix de routage a priori en certains sommets de l'arborescence. Les choix effectués ne garantissent pas un routage optimal et cette réduction de l'arborescence se ferait donc de façon heuristique.

Par exemple, lorsque se présente le choix de passer dans un coupleur ou non pour des fibres provenant de plusieurs endroits, on peut choisir d'éliminer certaines possibilités. L'idée de l'heuristique est de privilégier, d'une part les fibres les plus longues pour passer dans un coupleur, et d'autre part les fibres les plus courtes pour continuer sans passer dans un coupleur. Ce choix tend à équilibrer les distances entre OLT et clients (et donc le budget optique).

Il est possible d'élaguer l'arborescence selon ce principe pour les trois niveaux :

- Pour le premier niveau, dans le cas où il existe p chemins de longueurs respectives croissantes $l_1 \leq \dots \leq l_p$ permettant d'accéder au sommet i , où sont installés des coupleurs de niveau 1 et dont s'échappe des fibres de niveau 1.

- Lorsque la propriété suivante : $f_{ji}^1 \leq \sum_{l \neq i} f_{il}^1 - \sum_{l=1}^j f_{li}^1$ est vérifiée, aucune fibre issue du chemin passant par j n'alimentera un coupleur de niveau 1 situé en i .

- De façon symétrique, lorsque la propriété suivante : $f_{ji}^1 \leq z_i^1 - \sum_{l=j}^p f_{li}^1$ est vérifiée, toutes les fibres issues du chemin passant par j alimenteront un coupleur de niveau 1 situé en i . Dans ce cas, si des coupleurs de niveau 2 sont également installés en i , alors :

- Si la propriété suivante : $c^1 f_{ji}^1 \leq \sum_{l \neq i} f_{il}^2 - c^1 \sum_{l=1}^j f_{li}^1$

- Pour le second niveau, c'est un peu plus complexe car les coupleurs de niveau 2 peuvent être alimentés directement par des coupleurs de niveau 1 installés au même endroit. Plaçons nous donc dans le cas où il existe p chemins de longueurs respectives croissantes $l_1 \leq \dots \leq l_p$ permettant d'accéder au sommet i par des fibres de niveau 2.

- Lorsque la propriété suivante : $f_{ji}^2 \leq \sum_{l \neq i} f_{il}^2 - c^1 z_i^1$ est vérifiée, aucune fibre issue du chemin passant par j n'alimentera un coupleur de niveau 1 situé en i .

- Si dans la solution du problème de localisation de coupleurs, l'une ou les deux propriétés suivantes sont vérifiées

$$\sum_{j \neq i} f_{ij}^2 \geq c^1 z_i^1 \text{ et } \exists k \mid f_{ik}^2 > 0$$

ou

$$\sum_{j \neq i} f_{ij}^2 \geq c^1 z_i^1 \text{ et } \exists k \mid f_{ik}^2 > 0$$

alors il existe une solution où tous les coupleurs de niveau 2 installés en i sont alimentés par les coupleurs de niveau 1 installés eux-aussi en i . En particulier, lorsqu'il existe un j tel que $f_{ji}^2 > 0$, alors il existe une solution où aucune fibre venant de j n'alimente de coupleurs de niveau 2.

Démonstration

$$\sum_{j \neq i} f_{ji}^2 \quad \sum_{j \neq i} f_{ij}^2$$

$$\mathbf{z}_i^2 \leq \sum_{j \neq i} \mathbf{f}_{ji}^2 - \sum_{j \neq i} \mathbf{f}_{ij}^2 + c^1 \mathbf{z}_i^1$$

$$\sum_{j \neq i} \mathbf{f}_{ij}^2 - (c^1 \mathbf{z}_i^1 - \mathbf{z}_i^2) \leq \sum_{j \neq i} \mathbf{f}_{ji}^2$$

6.3.3 Première modélisation

Dans ce problème de choix de câblage il y a plusieurs décisions à prendre. Les premières concernent la répartition des fibres parmi les différents chemins représentés par l'arborescence. Ensuite, il faut affecter un câble approprié à chaque arc par lequel des fibres passent. Mais les décisions ne s'arrêtent pas là. À chaque changement de type de câble, il faut réaliser une jonction appelée épissure. Comme énoncé précédemment, une épissure dégrade légèrement la qualité du signal. On dit qu'elle consomme une partie du budget optique. Ce budget optique est essentiellement consommé par les coupleurs optiques, mais également par les épissures, la distance, et les différents connecteurs utilisés au niveau de l'OLT et du client. Pour chaque noeud de l'arborescence, tous ces facteurs sont connus sauf les épissures. Nous devons donc intégrer au modèle une contrainte limitant le nombre d'épissure entre les sommets de l'arborescence et la racine.

Lorsque les câbles incidents à un noeud sont différents des câbles en sortie, nous savons qu'une ou plusieurs épissures doivent être réalisées. Mais pas nécessairement sur tous les câbles en sortie. Sur l'exemple de la figure 6.14, un des trois câbles de taille 2 en sortie n'a pas besoin d'épissure : c'est le prolongement direct du câble de taille 2 en entrée.

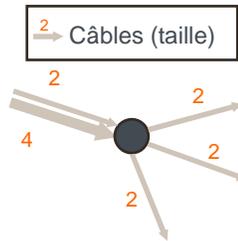


FIGURE 6.14 – Exemple de situation où il faut décider sur quels câbles les épissures sont pratiquées

Il nous faut donc décider sur quels câbles les épissures sont pratiquées. Les variables de décision pour notre problème sont donc les suivantes :

- \mathbf{x}_a : variables entières indiquant le nombre de fibres passant par l'arc a ;
- \mathbf{s}_a : variables entières indiquant le nombre de coupleurs de niveau 2 utilisés à l'extrémité terminale de l'arc a de niveau 1 ;
- \mathbf{g}_a^t : variables entières indiquant le nombre de câbles de type t installés sur l'arc a de l'arborescence ;
- \mathbf{e}_a : variables binaires indiquant si une épissure a été réalisée au départ de l'un des câbles installés sur l'arc a ;
- \mathbf{e}_a^t : variables entières indiquant le nombre de câbles de type t installés sur l'arc a qui débutent sur cet arc.

Les variables \mathbf{x}_a et \mathbf{s}_a permettent de décider quels chemins de l'arborescence seront empruntés et par des fibres. Il est aussi possible grâce à elles de savoir combien de coupleurs sont utilisés à l'extrémité terminale d'un arc a de l'arborescence. Le nombre de coupleurs de niveau k utilisés à l'extrémité terminale de l'arc a , de niveau k également, est donné par la formule :

$$\mathbf{x}_a - \sum_{u \in \Gamma^k(a)} \mathbf{x}_u \quad (6.41)$$

où $\Gamma^k(a)$ désigne les arcs de niveau k successeurs de a dans l'arborescence.

Les variables \mathbf{e}_a^t permettent de prendre en compte le coût des épissures et des raccordement de câbles dans l'objectif. Elles permettent aussi de calculer les variables \mathbf{e}_a qui interviennent dans les contraintes de budget optique limitant le nombre d'épissures entre l'OLT et les clients.

Le rôle des variables \mathbf{g}_a^t est quant lui très explicite.

Nous avons également besoin des notations suivantes :

- A est l'ensemble des arcs de l'arborescence ;
- A^k est l'ensemble des arcs de niveau k de l'arborescence ;
- C est l'ensemble des arcs pendants de l'arborescence ;
- T est l'ensemble des types de câble ;
- E^t est le coût d'une épissure pour raccorder un câble de type t ;
- G^t est le coût au mètre d'un câble de type t ;
- d_a est la longueur de l'arc a ;
- $budget_a$ nombre maximum d'épissures autorisé entre l'OLT et l'extrémité terminale de a .
- f_{ij}^k le nombre de fibres de niveau k installé sur l'arête (i, j) dans le graphe G du problème \mathcal{LCO} ;
- h_{aij}^k vaut 1 lorsque l'arc a correspond aux fibres de niveau k sur l'arête (i, j) dans le graphe G du problème \mathcal{LCO} , 0 sinon ;
- β_{ua} vaut 1 lorsque l'arc u fait partie du chemin entre l'OLT et l'extrémité terminale de l'arc a , 0 sinon ;

– M_a^t est une borne supérieure du nombre de câbles installés sur l'arc a . Elle peut

valoir par exemple
$$\left\lceil \frac{\sum_{ij} h_{aij} f_{ij}^k}{\min_t \{capa_t\}} \right\rceil.$$

6.3.3.1 Contraintes sur les chemins pris par les fibres et l'utilisation des coupleurs

Nous allons d'abord détailler les contraintes liées au choix de routage des fibres dans l'arborescence. Les variables \mathbf{x}_a et \mathbf{s}_a sont soumises aux contraintes suivantes :

$$\sum_{a \in A} \sum_{j \in V} h_{aji}^1 \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) = z_i^1 \quad \forall i \in V \quad (6.42)$$

$$\sum_{a \in A} \sum_{j \in V} \left[h_{aji}^2 \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) + h_{aji}^1 \mathbf{s}_a \right] = z_i^2 \quad \forall i \in V \quad (6.43)$$

$$\left(\mathbf{x}_a - \sum_{u \in \Gamma^1(a)} \mathbf{x}_u \right) c^1 - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \geq \mathbf{s}_a \quad \forall a \in A^1 \quad (6.44)$$

$$c^2 \mathbf{s}_a - \sum_{u \in \Gamma^3(a)} \mathbf{x}_u \geq 0 \quad \forall a \in A^1 \quad (6.45)$$

$$\left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) c^2 - \sum_{u \in \Gamma^3(a)} \mathbf{x}_u \geq 0 \quad \forall a \in A^2 \quad (6.46)$$

$$\left(\mathbf{x}_a - \sum_{u \in \Gamma^3(a)} \mathbf{x}_u \right) \geq 0 \quad \forall a \in A^3 \quad (6.47)$$

$$\sum_{a \in A} h_{aij}^k \mathbf{x}_a = f_{ij}^k \quad \forall (i, j) \in E \mid f_{ij}^k > 0, \forall k \quad (6.48)$$

Les contraintes (6.42) s'assurent que le nombre de coupleurs de niveau 1 utilisés par les sommets de l'arborescence correspondant au même sommet i du graphe G , est égal au nombre de coupleurs de niveau 1 installé en i . Les contraintes (6.43) s'assurent de la même chose mais pour les coupleurs de niveau 2.

Les contraintes (6.44) et (6.45) s'assurent du bon écoulement des fibres lors du passage sur un arc a de niveau 1, comme illustré dans la figure 6.15. Les contraintes (6.46) s'assurent du bon écoulement des fibres lors du passage sur un arc a de niveau 2, comme illustré dans la figure 6.15. Les contraintes (6.47) stipulent juste qu'il doit

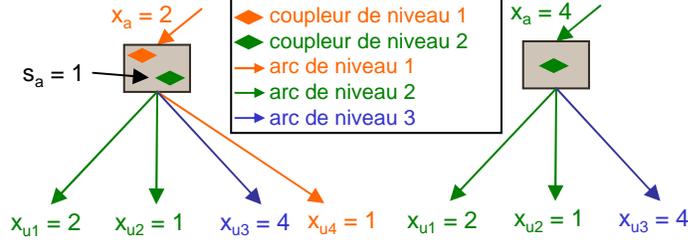


FIGURE 6.15 – Illustration des contraintes (6.44) et (6.45) d’une part, et (6.46) d’autre part, avec des coupleurs de capacité 4.

y avoir plus de fibres arrivant par un arc a de niveau 3 qu’il n’y en a qui repartent de cet arc. Enfin, les contraintes (6.48) s’assurent que le nombre de fibres passant par les arcs de l’arborescence correspond bien au nombre de fibres installées dans la solution de localisation des coupleurs.

En réalité, les contraintes (6.42) sont redondantes. Elles sont en effet impliquées par les contraintes (6.48) et les contraintes (6.1) du problème (\mathcal{LCO}) . En revanche, les contraintes (6.43) ne sont pas superflues. En effet, les contraintes (6.48), (6.44) et les contraintes (6.1) du problème (\mathcal{LCO}) impliquent seulement l’inégalité suivante :

$$\begin{aligned}
 \sum_{a \in A} \sum_{j \in V} \left[h_{aji}^2 \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) + h_{aji}^1 \mathbf{s}_a \right] &\leq \sum_{a \in A} \sum_{j \in V} \left[h_{aji}^2 \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) + h_{aji}^1 \left(\mathbf{x}_a - \sum_{u \in \Gamma^1(a)} \mathbf{x}_u \right) \right] c^1 - h_{aji}^1 z_i^1 \\
 &\leq \sum_{j \in V} \left[\sum_{a \in A} h_{aji}^2 \mathbf{x}_a - \sum_{a \in A} \left(h_{aji}^2 \sum_{u \in \Gamma^2(a)} \mathbf{x}_u + h_{aji}^1 \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) \right] + c^1 z_i^1 \\
 &\leq \sum_{j \in V} \left[f_{ji}^2 - \sum_{u \in A} h_{uij}^2 \mathbf{x}_u \right] + c^1 z_i^1 \\
 \sum_{a \in A} \sum_{j \in V} \left[h_{aji}^2 \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) + h_{aji}^1 \mathbf{s}_a \right] &\leq \sum_{j \in V} [f_{ji}^2 - f_{ij}^2] + c^1 z_i^1 \quad (\neq z_j^2)
 \end{aligned}$$

Si dans la solution de déploiement utilisée, la contrainte (6.2) pour i n’est pas saturée, alors la contrainte (6.43) peut donc se révéler utile. La figure illustre une situation où le passage des fibres dans l’arborescence ne respecte pas la contrainte (6.43).

6.3.3.2 Contraintes sur les épissures

Pour comprendre combien d’épissures sont nécessaires, il est important de noter qu’il est possible de faire du *piquage* sur les câbles. Le *piquage* est une technique consistant

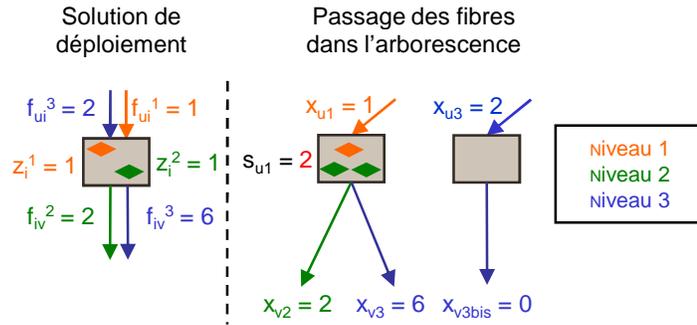


FIGURE 6.16 – Cas où le passage des fibres dans l'arborescence utilise trop de coupleurs de niveau 2

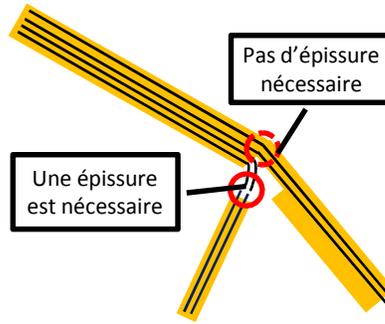


FIGURE 6.17 – Illustration du piquage

à extraire une partie des fibres d'un câble sans le couper. Une fenêtre est taillée dans la gaine du câble et les fibres sont prélevées à l'intérieur. Le câble sur lequel on a réalisé le piquage n'est pas coupé et continue sa route, mais une partie des fibres à l'intérieur ont été coupées et ne sont plus utilisables. Cette technique permet de réduire le nombre d'épissures nécessaires et est illustrée par la figure 6.17.

Le nombre d'épissures à pratiquer pour chaque type de câble est lié aux câbles installés sur un arc a de niveau k et sur ses successeurs de même niveau. S'il y a plus de câbles d'un type t installés sur les successeurs, qu'il n'y en a d'installés sur l'arc a lui-même, alors les nouveaux câbles ont subi une épissure. Cela se traduit par la relation suivante :

$$\max \left\{ \sum_{u \in \Gamma^k(a)} \mathbf{g}_u^t - \mathbf{g}_a^t ; 0 \right\} = \sum_{u \in \Gamma^k(a)} \mathbf{e}_u^t$$

Une épissure nécessite du matériel et de la main d'oeuvre. Elle a donc un coût qu'il

faut prendre en compte dans le modèle. Par conséquent, \mathbf{e}_a^t apparaît dans l'objectif avec un coefficient positif. Le modèle va donc chercher à minimiser le nombre de câbles coupés. On peut donc remplacer l'égalité précédente par la contrainte suivante :

$$\sum_{u \in \Gamma^k(a)} \mathbf{g}_u^t - \mathbf{g}_a^t \leq \sum_{u \in \Gamma^k(a)} \mathbf{e}_u^t \quad (6.49)$$

Il nous faut maintenant modéliser le lien entre les \mathbf{e}_a^t et \mathbf{e}_a . Il est possible d'utiliser les contraintes suivantes :

$$\mathbf{e}_a^t \leq \mathbf{g}_a^t \quad (6.50)$$

$$\mathbf{e}_a^t \leq M_a^t \mathbf{e}_a \quad (6.51)$$

La première inégalité (6.50) modélise le fait qu'on ne peut couper plus de câbles qu'il n'y en a d'installés. Dans la seconde inégalité (6.51), M_a^t est une borne supérieure du nombre de câbles de type t pouvant être installé sur l'arc a . L'inégalité (6.51) signifie que dès qu'on coupe un des câbles installés, nous devons considérer qu'une épissure a été réalisée sur tous les câbles de l'arc a . On effectue donc l'hypothèse simplificatrice suivante :

Propriété 6.8. *Les câbles installés sur un même arc de l'arborescence ont le même budget optique.*

Cette hypothèse restreint l'ensemble des solutions admissibles mais nous permet de compter facilement le nombre d'épissures entre un abonné et l'OLT. La contrainte de budget optique limitant les épissures entre l'OLT et l'extrémité terminale de l'arc a est la suivante :

$$\sum_{u \in A} \beta_{ua} \mathbf{e}_u \leq \text{budget}_a \quad (6.52)$$

Il est possible d'intégrer ces contraintes uniquement pour les arcs pendants de l'arborescence ($a \in C$), car cet ensemble de contraintes implique que le budget optique est respecté pour l'ensemble de l'arborescence.

6.3.3.3 Modèle

En se basant sur les contraintes exprimées précédemment, notre problème de câblage se modélise sous la forme du programme linéaire en nombre entiers suivant,

noté (CBL) :

$$\min_{\mathbf{g}_a^t, \mathbf{e}_a^t, \mathbf{e}_a, \mathbf{x}_a} \sum_{a \in A} d_a \sum_{t \in T} G^t \mathbf{g}_a^t + \left(\sum_{a \in A} \sum_{t \in T} E^t \mathbf{e}_a^t \right)$$

s.c. :

$$\sum_{a \in A} \sum_{i \in V} \left[h_{aij}^2 \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) + h_{aij}^1 \mathbf{s}_a \right] = z_j^2 \quad \forall j \in V \mid f_{ij}^2 > 0 \quad (6.43)$$

$$\left(\mathbf{x}_a - \sum_{u \in \Gamma^1(a)} \mathbf{x}_u \right) c^1 - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \geq \mathbf{s}_a \quad \forall a \in A^1 \quad (6.44)$$

$$c^2 \mathbf{s}_a - \sum_{u \in \Gamma^3(a)} \mathbf{x}_u \geq 0 \quad \forall a \in A^1 \quad (6.45)$$

$$\left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) c^2 - \sum_{u \in \Gamma^3(a)} \mathbf{x}_u \geq 0 \quad \forall a \in A^2 \quad (6.46)$$

$$\left(\mathbf{x}_a - \sum_{u \in \Gamma^3(a)} \mathbf{x}_u \right) \geq 0 \quad \forall a \in A^3 \quad (6.47)$$

$$\sum_{a \in A} h_{aij}^k \mathbf{x}_a = f_{ij}^k \quad \forall i, j \in V \mid f_{ij}^k > 0, \forall k \quad (6.48)$$

$$\sum_{u \in \Gamma^k(a)} \mathbf{g}_u^t - \mathbf{g}_a^t \leq \sum_{u \in \Gamma^k(a)} \mathbf{e}_u^t \quad \forall a \in A^k, \forall t = 1 \dots T, \forall k = 1, 2, 3 \quad (6.49)$$

$$\mathbf{e}_a^t \leq \mathbf{g}_a^t \quad \forall a \in A, \forall t = 1 \dots T \quad (6.50)$$

$$\mathbf{e}_a^t \leq M_a^t \mathbf{e}_a \quad \forall a \in A, \forall t = 1 \dots T \quad (6.51)$$

$$\sum_{t \in T} \text{capa}_t \mathbf{g}_a^t \geq \mathbf{x}_a \quad \forall a \quad (6.53)$$

$$\sum_{u \in A} \beta_{ua} \mathbf{e}_u \leq \text{budget}_a \quad \forall a \in C \quad (6.52)$$

$$\mathbf{e}_a^t, \mathbf{g}_a^t \in \mathbb{N}; \mathbf{e}_a \in \{0, 1\}$$

Les seules contraintes qui n'ont pas été expliquées précédemment sont les contraintes (6.53). Les contraintes (6.53) s'assurent que les câbles installés sur l'arc a ont une capacité suffisante pour le nombre de fibres passant sur l'arc a .

Il est tout à fait possible d'avoir des types de câbles différents pour les différents niveaux de fibres. Dans ce cas, nous noterons T^k l'ensemble des types de câble autorisés pour le niveau k . Établir une telle différence peut permettre de réduire le nombre de variables et de contraintes présentes dans le modèle.

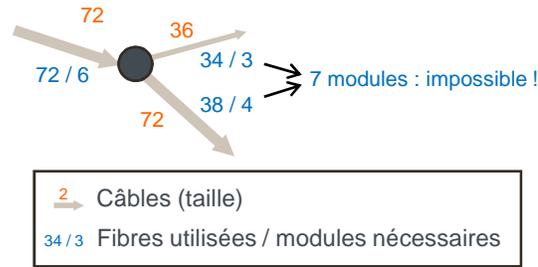


FIGURE 6.18 – Illustration d’une solution irréalisable à cause de la modularité des câbles (ici 12)

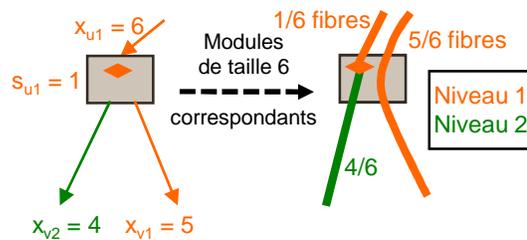


FIGURE 6.19 – Exemple de cas où l’indivisibilité des modules amène à devoir en installer deux au lieu d’un seul

6.3.4 Problème de la modularité des câbles

Dans les gros câbles, les fibres ne sont pas accessibles une à une mais par module de plusieurs fibres. En cas de piquage, c’est le module entier qui est extrait. Les fibres non-utilisées d’un module piqué ne sont donc plus disponibles dans le câble. Ne pas tenir compte de ce phénomène peut amener à proposer des solutions non opérationnelles. Par exemple, dans le cas illustré par la figure 6.18.

Le module est l’unité de base d’une solution opérationnelle de câblage. Cela signifie que les fibres d’un même module sont indissociables, même en cas d’épissure. Par exemple, si une fibre d’un module est utilisée pour alimenter un coupleur dans une chambre i , alors le reste des fibres de ce module s’arrêtent elle aussi dans cette chambre. Elles ne peuvent être utilisées que pour alimenter d’autres coupleurs de même niveau dans cette même chambre. Cela peut amener à devoir installer plus de modules que le nombre requis par leur capacité, comme l’illustre la figure 6.19.

Nous noterons \mathbf{m}_a le nombre de modules installés sur l’arc a , et mod^k le nombre de fibres optiques par module de niveau k . En effet, s’il est possible d’avoir différents types de câbles pour chaque niveau de fibre, il est également possible d’avoir différentes tailles de modules selon le niveau. L’introduction des modules nécessite le

rajout des contraintes suivantes :

$$\text{mod}^1 \left(\mathbf{m}_a - \sum_{u \in \Gamma^1(a)} \mathbf{m}_u \right) \geq \left(\mathbf{x}_a - \sum_{u \in \Gamma^1(a)} \mathbf{x}_u \right) \quad \forall a \in A^1 \quad (6.54)$$

$$\text{mod}^2 \left(\mathbf{m}_a - \sum_{u \in \Gamma^2(a)} \mathbf{m}_u \right) \geq \left(\mathbf{x}_a - \sum_{u \in \Gamma^2(a)} \mathbf{x}_u \right) \quad \forall a \in A^2 \quad (6.55)$$

$$\left(\mathbf{m}_a - \sum_{u \in \Gamma^3(a)} \mathbf{m}_u \right) \geq 0 \quad \forall a \in A^3 \quad (6.56)$$

$$\text{mod}^k \mathbf{m}_a \geq \mathbf{x}_a \quad \forall a \in A^k \quad (6.57)$$

et les contraintes (6.53) sont remplacées par :

$$\sum_{t \in T^k} \text{capamod}_t \mathbf{g}_a^t \geq \mathbf{m}_a \quad \forall a \in A^k \quad (6.53)$$

où capamod_t représente le nombre de modules de niveau k contenu dans un câble de type $t \in T^k$.

Les contraintes (6.54) s'assurent que les coupleurs de niveau 1 utilisés à l'extrémité terminale de l'arc a sont desservis par des modules spécifiques qui s'arrêtent à cette extrémité et ne continuent pas plus loin. Le principe est le même pour les contraintes (6.55) et les coupleurs de niveau 2. Les contraintes (6.56) s'assurent qu'il y a plus de modules installés sur l'arc a que sur ces successeurs. Il est important de noter que les contraintes (6.54) et (6.55) impliquent l'équivalent des contraintes (6.56) pour les arcs de niveau 1 et 2.

Les contraintes (6.57) indiquent simplement qu'il faut un nombre de modules suffisant pour satisfaire le besoin en fibre.

6.3.5 Contraintes sur le nombre de câble

Les règles de tubage des conduites, dans lesquelles passent les câbles, peuvent entraîner une limitation du nombre de câbles sur certaines arêtes du graphe G . Supposons que les types de câble sont ordonnés par diamètres croissants. En notant b_{ij}^t le nombre de câbles de type t pouvant passer sur l'arête (i, j) , et γ^{st} le nombre de câble de type

s occupant la place d'un seul câble de type t , on obtient les contraintes suivantes :

$$\begin{aligned} \sum_{a \in A} \sum_{k=1}^3 h_{aij}^k \mathbf{g}_a^T &\leq b_{ij}^T & \forall (i, j) \in E \mid \sum_{k=1}^3 f_{ij}^k > 0 \\ \sum_{a \in A} \sum_{k=1}^3 h_{aij}^k \left(\mathbf{g}_a^{T-1} + \gamma^{(T-1)T} \mathbf{g}_a^T \right) &\leq b_{ij}^{T-1} & \forall (i, j) \in E \mid \sum_{k=1}^3 f_{ij}^k > 0 \\ &\vdots \\ \sum_{a \in A} \sum_{k=1}^3 h_{aij}^k \left(\mathbf{g}_a^1 + \sum_{t=2}^T \gamma^{1t} \mathbf{g}_a^t \right) &\leq b_{ij}^1 & \forall (i, j) \in E \mid \sum_{k=1}^3 f_{ij}^k > 0 \end{aligned}$$

Ces contraintes sont plus précises que celles présentes dans le problème de localisation de coupleurs qui portaient uniquement sur le nombre total de fibres pouvant passer sur une arête. Cependant, la taille du modèle étant déjà importante, l'opportunité d'encombrer le modèle en rajoutant ces contraintes doit être discutée.

6.3.6 Modélisation alternative des câbles

Une autre façon de modéliser le problème consiste à créer une variable \mathbf{g}_l par câble. La solution obtenue est alors plus facile à traduire en solution opérationnelle. Le prix du câble l , noté P_l serait alors :

$$P_l = E^{\theta_l} + \sum_{a \in A} \eta_{al} d_a G^{\theta_l}$$

où :

- θ_l est le type du câble l ;
- η_{al} vaut 1 si le câble l passe par l'arc a , 0 sinon ;
- δ_{al} vaut 1 si le câble l débute par l'arc a .

La fonction objectif devient alors : $\sum_{l \in L} P_l \mathbf{g}_l$, où L est l'ensemble des câbles, et les contraintes (6.53), (6.49), (6.50), et (6.51) sont remplacées par les contraintes suivantes :

$$\sum_{l \in L} \eta_{al} \text{cap}_{al} \mathbf{g}_l \geq \mathbf{x}_a \quad \forall a \in A \quad (6.58)$$

$$\sum_{l \in L} \delta_{al} \mathbf{g}_l \leq M_a \mathbf{e}_a \quad \forall a \in A \quad (6.59)$$

où M_a est une borne sur le nombre de câble installé sur l'arc a .

Pour résoudre le problème ainsi obtenu, il faut alors mettre en place une procédure de Branch&Price . Le coût réduit de la variable \mathbf{g}_l est :

$$\begin{aligned}\bar{P}_l &= P_l - \sum_{a \in A} \eta_{al} \text{capa}_l \lambda_a + \sum_{a \in A} (\eta_{al} - \eta_{\Gamma^-(a)l}) \mu_a \\ &= \sum_{a \in A} \delta_{al} E^{\theta_l} + \sum_{a \in A} \eta_{al} d_a G^{\theta_l} - \sum_{a \in A} \eta_{al} \text{capa}_l \lambda_a + \sum_{a \in A} \delta_{al} \mu_a \\ &= \sum_{a \in A} \delta_{al} E^{\theta_l} + \sum_{a \in A} \eta_{al} (d_a G^{\theta_l} - \text{capa}_l \lambda_a) + \sum_{a \in A} \delta_{al} \mu_a\end{aligned}$$

où les λ_a sont les variables duales associées aux contraintes (6.58), et où les μ_a sont les variables duales associées aux contraintes (6.59).

Trouver le câble de type t ayant le coût réduit minimum revient à un problème de plus court chemin dans la forêt des différents niveaux de l'arborescence munie de la valuation $(d_a G^t - \text{capa}_l \lambda_a)$, et à laquelle on a rajouté une source reliée à tous les sommets par des arcs de poids $\mu_a + E^t$ (ou 0 pour les sommets racines d'un des arbres de la forêt), ainsi qu'un puits relié à tous les sommets par des arcs de poids nul. Il nous faut résoudre ce problème pour tous les types de câble et comparer les coûts réduits des câbles obtenus. Le sous-problème est donc polynomial en la taille de l'arborescence et le nombre de types de câble.

Si l'on enlève le puits on obtient un arbre ayant moins de deux fois le nombre d'arcs de l'arborescence. Plus exactement, un arc est rajouté entre la source et chaque sommet à l'exception des feuilles (inutile de faire débiter un câble au niveau d'une feuille). Le graphe du sous-problème a donc $2 \times |A| - |C|$ arcs. La figure 6.20 montre le graphe correspondant à l'arborescence de la figure 6.13.

Si l'on veut rajouter des contraintes de longueur sur les câbles, on obtient alors un problème de calcul de plus court chemin contraint. Mais le sous-problème reste polynomial car le nombre de chemins à énumérer est borné polynomialement. En effet, dans le pire des cas un sommet peut être atteint par $|A|$ chemins différents. Le nombre de chemins à évaluer est donc inférieur à $|A|(|A| + 1)$.

Une formulation plus serrée pour le problème consistera à remplacer les contraintes (6.59) par les contraintes suivantes :

$$\delta_{al} \mathbf{g}_l \leq \mathbf{e}_a \forall a \in A, \forall l \in L \quad (6.60)$$

L'impact de ce changement sur le sous-problème reste à étudier.

6.3.7 Reconstruction d'une solution opérationnelle

La solution du modèle (CBL) nous donne le nombre de câbles de chaque type installés sur chaque arc de l'arborescence et donc sur chaque arête de voirie. Elle nous donne

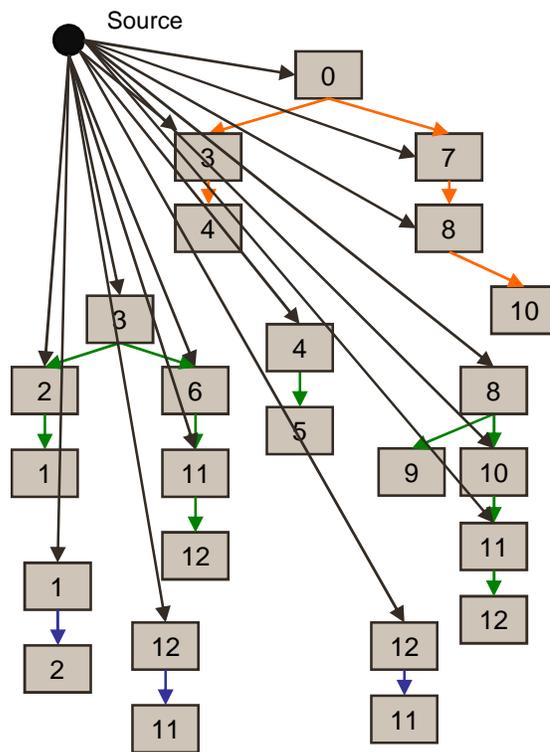


FIGURE 6.20 – Exemple de graphe du sous-problème

aussi le détails du routage des fibres (et éventuellement des modules de fibres) allant de l'OLT aux clients.

Il nous faut maintenant reconstruire à partir de la solution de câblage une solution contenant toutes les informations nécessaires pour une exploitation par l'opérationnel. Les câbles dans la solution du problème de câblage (dans la première modélisation) ne sont pas identifiés individuellement. Or il faut qu'en cas d'épissure, on sache précisément quels câbles sont utilisés. Ils nous reste donc quelques décisions à prendre, bien qu'elles n'influent pas sur le coût du déploiement.

Nous voulons donc obtenir ces données dans la forme suivante :

- Pour chaque arête du graphe G (le graphe des conduites existantes ou de la voirie) : la référence individuelle de chaque câble installé le long de cette arête ;
- Pour chaque sommet du graphe G (chambres en béton où il est possible d'installer des coupleurs et de pratiquer des épissures) :
 - le détails des coupleurs installés : sa référence, son niveau, son type, la référence du câble utilisé en entrée, la liste des références des câbles utilisés en sortie.
 - le détails des épissures effectuées sur le noeud : la référence de l'épissure, la référence du coupleur ou du câble en amont, la référence du coupleur ou du câble en aval, l'indication qu'il y a piquage ou non ;
- La liste des câbles installés avec pour chacun d'entre eux : sa longueur, la liste des sommets parcourus, la référence du câble ou du coupleur (ou bien encore de l'OLT) auquel il est raccordé en amont, la référence du câble ou du coupleur (ou bien encore du PRI) auquel il est raccordé en aval.

Ces données peuvent être renseignées à parcourant la solution du problème d'affectation de câblage. À chaque fois qu'un câble ou un coupleur est rencontré pour la première fois, sa référence unique est créée.

Mise à part ces informations nécessaires pour pouvoir déployer la solution, il peut être intéressant d'obtenir certains indicateurs comme la distance moyenne entre un client et son coupleur de second niveau. Ces indicateurs sont nécessaires pour l'évaluation des règles d'ingénierie actuelles.

6.4 Conclusion et perspectives

Les particularités des réseaux accès FTTH avec une architecture GPON font de l'optimisation de leur déploiement un problème complexe. Cette complexité se retrouve tant sur le plan théorique que sur le plan pratique. En effet, les données sont souvent difficilement accessibles, et l'analyse des contraintes et des enjeux n'est pas évidente.

Cette complexité se retrouve dans les modèles proposés pour nos deux sous-problèmes : le problème de localisation de coupleurs optiques et celui du choix du câblage. Faire en sorte que ces modèles correspondent exactement à la situation opérationnelle est

extrêmement difficile. Mais pour que ces modèles soient pertinents pour évaluer les règles d'ingénierie, ils doivent s'approcher de cette réalité opérationnelle le plus possible. C'est ainsi que la prise en compte du dimensionnement des coupleurs pour le problème de localisation de coupleurs, ou celle de la modularité des câbles pour le problème du choix du câblage, se révèlent indispensables même si elles complexifient les modèles.

Pour le problème de localisation de coupleurs optiques, les tests ont montré qu'il était possible de trouver de bonnes solutions en un temps raisonnable. L'instance la moins dense semble plus difficile, mais les inégalités valides rajoutées au problème ont permis de réduire l'écart à l'optimal à un niveau acceptable.

Plusieurs indicateurs ont été extraits de ces solutions : deux indicateurs sur l'usage qui est fait des coupleurs et un indicateur sur la façon dont sont satisfaites les demandes. Ces indicateurs donnent un premier aperçu de la pertinence des solutions proposées et de certaines règles d'ingénierie.

Bien que nous obtenions de bonnes solutions, les méthodes de résolution peuvent encore être améliorées. Par exemple, avec le développement d'heuristiques. Une plus grande efficacité de ces méthodes permettrait d'améliorer encore la pertinence du modèle, notamment en prenant en compte la portée maximale au sein du problème de localisation de coupleurs. En effet, à l'heure actuelle, la contrainte d'éligibilité n'est prise en compte que dans le problème de câblage. Cela peut causer des problèmes d'infaisabilité du câblage, si la solution de localisation des coupleurs ne permet pas de respecter cette contrainte d'éligibilité.

Une prise en compte du routage du trafic associé à chaque client au sein du problème de localisation de coupleur est également à étudier, car cela simplifiera grandement la construction de l'arborescence utilisée dans le problème de câblage.

Concernant le problème de câblage, les modèles doivent encore être testés. La taille de l'arborescence décrivant la solution de déploiement est un enjeu majeur pour que ces modèles soient utilisables. Dans nos premières expérimentations la taille de cette arborescence dépendait grandement de la qualité des solutions du problème de localisation de coupleurs. C'est pourquoi nous avons focalisé nos efforts sur ce premier problème, mettant de côté les tests des modèles du second. Une des perspectives majeures à court terme est donc le test de ces modèles sur les instances réelles, afin d'arriver enfin à une solution de déploiement détaillée.

Conclusion

Dans cette thèse, nous avons abordé deux applications de problèmes d'optimisation dans les réseaux d'accès de télécommunications. Ces réseaux d'accès se présentent sous forme de réseaux hiérarchiques, avec au plus haut niveau un concentrateur central, au plus bas les clients, et entre les deux un ou plusieurs niveaux d'équipements intermédiaires.

La première application consiste à optimiser le nombre et le placement des sous-répartiteurs dans l'accès du Réseau Téléphonique Commuté. Cette optimisation se fait sur un territoire vierge. C'est à dire sur un territoire sans infrastructure existante où il n'y a pas d'emplacements potentiels pour les équipements, et où les liaisons entre les éléments se font en ligne droite. Les équipements intermédiaires sont considérés sans capacité. Ces simplifications permettent de résoudre des tailles d'instances correspondant au territoire associé à un concentrateur central dans les instances réelles. C'est à dire des instances avec plusieurs milliers de clients.

Pour cette première application, nous avons proposé et testé différentes heuristiques de résolution fournissant différents compromis entre qualité de la solution et temps de calcul nécessaire. Une des ces heuristiques est adaptée d'une heuristique multi-phase existante pour un problème proche de localisation, le problème de Weber multi-source. Les autres proposent de se servir conjointement des méthodes de recherche opérationnelle et d'un modèle de géométrie stochastique existant pour ce problème. La qualité des solutions de ces heuristiques a pu être évaluée grâce à une méthode de génération de colonnes que nous avons développée pour cette application.

Les différentes méthodes ont été testées sur un ensemble d'instances artificielles de grande taille, ainsi que sur deux instances issues de données réelles. La meilleure heuristique est l'heuristique multi-phase, qui apporte de très bonnes solutions en un temps raisonnable pour les instances présentant entre 1000 et 2000 clients. Pour les instances plus grandes, le temps de calcul devient prohibitif, et il faut alors utiliser les heuristiques hybrides utilisant le modèle de géométrie stochastique. Ces heuristiques proposent des solutions d'une qualité légèrement inférieure mais peuvent résoudre les instances les plus grandes avec près de 3000 clients.

Notre objectif qui était de proposer un éventail de méthodes offrant des solutions de

qualité en un temps raisonnable pour différentes taille d'instances réelles est donc atteint.

Plusieurs aspects de nos travaux sur l'optimisation du réseau d'accès du RTC peuvent dépasser le cadre de cette application. C'est le cas notamment de la méthode de génération de colonnes que nous avons développée. Notre problème présentant de gros problèmes de dégénérescence et nos tailles d'instance étant très élevées, nous avons mis l'accent sur la stabilisation. Pour cela, nous avons développé une méthode de génération de colonnes basée sur un algorithme de plans coupants central, auquel nous avons adapté des techniques de stabilisation classiques. Nous avons comparé cette nouvelle méthode à une génération de colonnes classique stabilisée pour notre problème. Sur les instances de grande taille, l'efficacité de notre nouvelle méthode est manifeste. Elle se montre systématiquement plus rapide que la génération de colonnes classique et son temps de calcul moyen est entre cinq et six fois inférieur. Nous n'avons pas eu l'occasion de tester cette nouvelle méthode sur d'autres problèmes au cours de cette thèse, mais nous avons bon espoir que son efficacité soit montrée à l'avenir sur d'autres applications souffrant de problèmes de stabilisation.

Le problème d'optimisation de l'accès du RTC étant proche de la variante du problème de Weber multi-source avec nombre d'équipements inconnus, nous y avons adapté les heuristiques à base de géométrie stochastique, ainsi que notre génération de colonnes. La génération de colonnes nous a ainsi permis d'évaluer la qualité de différentes méthodes de résolution pour ce problème difficile sur quelques instances de la littérature ainsi que sur des instances générées aléatoirement. La principale conclusion est l'excellente qualité de l'heuristique multi-phase développée pour ce problème par [7].

La deuxième application abordée par cette thèse est l'optimisation du déploiement d'un réseaux d'accès FTTH, en présence d'une infrastructure existante. Cette application répond au besoin actuel des opérateurs qui déploient des réseaux en fibre optique pour répondre à l'augmentation du trafic et pour pouvoir proposer de nouveaux services. Les coûts de génie civil importants nécessaires lors du déploiement de nouveaux réseaux poussent à vouloir réutiliser au maximum les infrastructures existantes comme les conduites libres déployées pour d'autres réseaux, ou encore les égouts.

Les particularités de l'architecture considérée, notamment les deux niveaux de coupleurs optiques qui permettent de « mutualiser » les fibres optiques entre le concentrateur central et les clients. Nous proposons donc dans cette thèse un ensemble de modèles pour deux problématiques liées au déploiement : la localisation des coupleurs optiques d'une part, et le choix du câblage d'autre part. Ces modèles ont pour but de proposer des solutions de référence permettant d'évaluer les règles d'ingénierie mises en place pour le déploiement sur le terrain.

Nous avons rencontré de nombreuses difficultés lors de la modélisation de ces pro-

blèmes, notamment pour obtenir les informations nécessaires : données des instances, coûts du réseau, contraintes et règles existantes, ... Cela explique la variété des modélisations proposées et le fait que les tests ne sont effectués que sur un petit nombre d'instances. Nous avons concentré nos efforts sur le problème de localisation de coupleurs car ce sont ses solutions qui doivent servir de base au problème de choix du câblage.

Nous avons pu tester différentes versions de notre problème de localisation de coupleurs avec un solveur de programmation linéaire en nombre entiers sur quelques instances réelles. Les problèmes ne sont pas résolus de façon exacte et une limite de temps est imposée au solveur. Des techniques de réduction sont utilisées pour réduire la taille des instances sans influencer sur la solution optimale.

Les résultats montrent que la qualité des solutions dépend de la topologie des instances et des différents aspects pris en compte ou non dans les modèles. Notre objectif étant de rapprocher le plus possible les modèles de la réalité opérationnelle, nous avons proposé une famille d'inégalités valides pour ce problème et l'avons implémentée dans un branch-&-cut. Cela nous a permis d'améliorer sensiblement la qualité des solutions dans certains cas.

Ces premières solutions peuvent fournir des indications sur la validité de certaines règles d'ingénierie, notamment la localisation du second niveau de coupleur ou l'impact des taux de dimensionnement des coupleurs. Mais pour une évaluation complète des règles d'ingénierie, d'autres facteurs doivent encore être intégrés aux modèles, et des progrès doivent donc être réalisés sur les méthodes de résolution.

Une autre perspective à court terme est le test des modèles de choix de câblage à partir des solutions de qualité obtenue pour la localisation de coupleurs. La résolution de ces modèles devrait mener à des solutions de déploiements complètes et opérationnelles, mais leur difficulté est encore inconnue.

Bibliographie

- [1] BACCELLI F, ZUYEV S. *Poisson-Voronoi spanning trees with applications to the optimization of communication networks*. Operations Research 1999 ; 47(4) ; 619-631.
- [2] J. E. BEASLEY, *Lagrangean heuristics for location problems*, European Journal of Operational Research, 65, 383-399, 1993
- [3] W. BEN-AMEUR et J. NETO, *Acceleration of Cutting-Plane and Column Generation Algorithms : Applications to Network Design*, Networks, Vol. 49, 3-17, 2007
- [4] B. BETRÒ, *An accelerated central cutting plane algorithm for linear semi-infinite programming*, Mathematical Programming, Vol.101, 479-495, 2004.
- [5] D. BIENSTOCK et O. GÜNLÜCK, *Capacitated network design - polyhedral structure and computation*, INFORMS Journal on Computing, 8, 243-259, 1996
- [6] J. BRIMBERG, P. HANSEN, N. MLADENović, et E. D. TAILLARD, *Improvements and Comparison of Heuristics for Solving the Uncapacitated Multisource Weber Problem*, Operations Research, 48, 444-460, 2000
- [7] J. BRIMBERG, N. MLADENOVIC, S. SALHI, *The multi-source Weber problem with constant opening cost*, Journal of the Operational Research Society, 55, 640-646, 2004.
- [8] T. CARPENTER et H. LUSS, *Telecommunication Access Network Design*, dans : M. G. C. RESENDE et P. M. PARDALOS (Eds), Handbook of Optimization in Telecommunication, 313-340, 2006
- [9] E. CARRIZOSA et A. M. RODRIGUEZ-CHIA, *Weber problems with alternative transportation systems*, European Journal of Operational Research, 97, 87-93, 1997
- [10] COOPER L. *Heuristic Methods for Location-Allocation Problems*. SIAM Reviews 1964 ; 6 ; 37-53.

- [11] G. CORNUÉJOLS, G.L. NEMHAUSER, L.A. WOLSEY, *The uncapacitated facility location problem*, dans : P.B. MIRCHANDIANI et R.L. FRANCIS (Eds), *Discrete Location Theory*, 119-171, 1990
- [12] F. R. B. CRUZ, G. R. MATEUS , J. MACGREGOR SMITH, *A Branch-and-Bound Algorithm to Solve a Multi-level Network Optimization Problem*, *Journal of Mathematical Modelling and Algorithms*, 2, 37-56, 2003
- [13] J. DESROSIERS et M. E. LÜBBECKE, *A Primer in Column Generation*, dans : G. DESAULNIERS, J. DESROSIERS et M. M. SOLOMON (Eds), *Column Generation*, 1-32, 2005
- [14] Z. DREZNER, K. KLAMROTH, A. SCHÖBEL et G. O. WESOLOWSKY, *The Weber Problem*, dans : Z. DREZNER et H. HAMACHER (Eds), *Facility Location : Applications and Theory*, 1-36, 2002
- [15] Z. DREZNER, A. MEHREZ, G. O. WESOLOWSKY, *The Facility Location Problem with Limited Distances*, *Transportation Science*, Vol.25, No.3, 183-187, 1991.
- [16] S. EILON, C. D. T. WATSON-GANDY, et N. CHRISTOFIDES (1971), *Distribution Management : Mathematical Modelling and Practical Analysis*, Hafner, New York, 1971
- [17] J. ELZINGA et T. G. MOORE, *A central cutting plane algorithm for the convex programming problem*, *Mathematical Programming*, Vol.8, 134-145, 1975.
- [18] E. GOURDIN, M. LABBÉ, et H. YAMAN, *Telecommunication and Location*, dans : Z. DREZNER et H. HAMACHER (Eds), *Facility Location : Applications and Theory*, 275-305, 2002
- [19] P. HANSEN, J. BRIMBERG, D. UROŠEVIĆ, N. MLADENović, *Primal-Dual Variable Neighborhood Search for the Simple Plant Location Problem*, *INFORMS Journal on Computing*, Vol. 19 , No. 4, 2007
- [20] P. HANSEN, N. MLADENOVIC et E. TAILLARD, *Heuristic Solution of the Multisource Weber Problem as a p-Median Problem*, *Operations Research Letters*, 22, 55-62, 1998
- [21] T. Koch et A. Martin, *Solving Steiner Tree Problems in Graphs to Optimality*, *Networks*, Vol. 32, No. 3, 207-232, 1998
- [22] M. KÖRKEL, *On the exact solution of large-scale simple plant location problems*, *European Journal of Operational Research*, 39, 157-173, 1989
- [23] S. KRAU, *Extensions du problème de Weber*, PhD Thesis, 1997.
- [24] Y. LEE, Y. KIM, J. HAN, *FTTH-PON Splitter Location-Allocation Problem*, *Proceedings of the eighth INFORMS Telecommunications Conference*, 2006
- [25] J. LI et G. SHEN, *Cost Minimization Planning for Greenfield Passive Optical Networks*, *Journal of Optical Communications and Networking*, Vol. 1, No. 1, 17-29, 2009

- [26] M. E. LÜBBECKE, J. DESROSIERS, *Selected Topics in Column Generation*, Operations Research, Vol. 53, No. 6, 1007-1023, 2005
- [27] J. E. KELLEY, *The cutting-plane method for solving convex programs*, Journal of the Society for Industrial and Applied Mathematics, Vol.8, No. 4, 703-712, 1960
- [28] G. R. MATEUS, F. R. B. CRUZ, H. P. L. LUNA, *An Algorithm for Hierarchical Network Design*, Location Science, Vol. 2, No. 3, 149-164, 1994
- [29] G. R. MATEUS et Z. K. G. PATROCÍNIO JR., *Optimization Issues in Distribution Network Design*, dans : M. G. C. RESENDE et P. M. PARDALOS (Eds), Handbook of Optimization in Telecommunication, 313-340, 2006
- [30] N. MEGIDDO et K. J. SUPOWIT, *On the Complexity of Some Common Geometric Location Problems*, SIAM Journal on Computing, 13, 182-196, 1984
- [31] P. J. NEAME, *Nonsmooth dual methods in integer programming*, PhD Thesis, University of Melbourne, 1999
- [32] A. OUOROU, *A Proximal Cutting Plane Method Using Chebychev Center for Nonsmooth Convex Optimization*, Mathematical Programming, Ser. A, 119, 239-271, 2009
- [33] R. A. PATTERSON et E. ROLLAND, *Hybrid Fiber Coaxial Network Design*, Operations Research, Vol. 50, No. 3, 538-551, 2002
- [34] B. PFEIFFER et K. KLAMROTH, *A unified model for Weber problems with continuous and network distances*, Computers & Operations Research, Vol. 35, No. 2, 312-326, 2008
- [35] K.F. POON, D.B. MORTIMORE, J. MELLIS, *Designing optimal FTTH and PON networks using new automatic methods*, The 2nd Institution of Engineering and Technology International Conference on Access Technologies, 2006.
- [36] C. D. RANDAZZO et H. P. L. LUNA, *A Comparison of Optimal Methods for Local Access Uncapacitated Network Design*, Annals of Operations Research, 106, 263-286, 2001
- [37] C. D. RANDAZZO, H. P. L. LUNA, P. MAHEY, *Benders Decomposition for Local Access Network Design with Two Technologies*, Discrete Mathematics and Theoretical Computer Science, 4, 235-246, 2001
- [38] G. REINELT, TSPLIB - a traveling salesman problem library, <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>
- [39] G. RIGHINI et L. ZANIBONI, *A branch-&-price algorithm for the multi-source Weber problem*, Int. J. Operational Research, Vol.2, No.2, 188-207, 2007.
- [40] K. E. ROSING, *An optimal method for solving the (generalized) multi-Weber problem*, European Journal of Operational Research, 58, 414-426, 1992

- [41] L-M. ROUSSEAU, M. GENDREAU, D. FEILLET, *Interior point stabilization for column generation*, Operations Research Letters, Vol. 35, 660-668, 2007
- [42] D. M. RYAN et B. A. FOSTER, *An integer programming approach to scheduling*, dans : A. WREN (Ed), Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling, North Holland, Amsterdam, 269-280, 1981
- [43] F. S. SALMAN, R. RAVI, et J. N. HOOKER, *Solving the Capacitated Local Access Network Design Problem*, INFORMS Journal on Computing, Vol. 20, No. 2, 243-254, 2008
- [44] H. D. SHERALI, Y. LEE, T. PARK, *New modeling approaches for the design of local access transport area networks*, European Journal of Operational Research, Vol. 127, 94-108, 2000
- [45] F. VANDERBECK, *Implementing Mixed Integer Column Generation*, dans : G. DESAULNIERS, J. DESROSIERS et M. M. SOLOMON (Eds), Column Generation, 331-358, 2005
- [46] L. A. WOLSEY, *Integer Programming*, John Wiley & Sons, New York, 1998