

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene
Faculté de Mathématiques



MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER :

En : MATHEMATIQUES

Spécialité Recherche Opérationnelle : *Mathématiques Discrètes et Optimisation*

Par : Chafia BOUGHANI

Thème

*Approche polyédrale de quelques problèmes
d'optimisation combinatoire*

Soutenu le 25/01/2009, devant le jury composé de :

M ^r M. ABBAS, Professeur à l'USTHB	Président
M ^r M. AÏDER, Professeur à l'USTHB	Directeur de thèse
M ^r A. SEMRI, Maître de conférence à l'USTHB	Examineur
M ^r D. CHAABANE, Maître de conférence à l'USTHB	Examineur

Remerciements

Au terme de ce travail, il est agréable d'adresser mon remerciement à tous ceux qui m'ont prêté main forte pour sa réalisation.

En premier lieu, j'exprime ma profonde et respectueuse gratitude à Monsieur *ADER* Méziane, Professeur à l'Université des Sciences et de la Technologie Houari Boumediene pour l'honneur qu'il m'a fait de bien vouloir encadrer mon travail et pour l'intérêt et l'enthousiasme manifestés. Je tiens également à le remercier pour la sympathie et de me diriger patiemment.

J'exprime mes sincères remerciements à Monsieur Moncef *ABBAS*, Professeur à l'Université des Sciences et de la Technologie Houari Boumediene, pour l'honneur et le plaisir qu'il m'a fait en acceptant de présider ce jury.

Mes remerciements vont également à Monsieur Djamel *CHAABANE*, Maître de conférence à l'Université des Sciences et de la Technologie Houari Boumediene, Monsieur Ahmed *SEMRJ*, Maître de conférence à l'Université des Sciences et de la Technologie Houari Boumediene, qui ont bien voulu examiner ce mémoire et participer au jury.

Mes remerciements s'adressent à ma famille.

Je remercie tous mes amis pour l'affectueuse amitié dont ils ont toujours fait preuve.

Dédicace



A Dieu, source de toute vie et de tout amour,

A ma famille qui m'a permis de devenir ce que je suis aujourd'hui,

A mes amis qui sont autant de rayons de soleil dans ma vie.



Chafia BOUGHANJ



TABLE DES MATIÈRES

<i>Table des matières</i>	<i>i</i>
<i>Table des figures</i>	<i>iii</i>
<i>Liste des tableaux</i>	<i>iv</i>
<i>Liste des algorithmes</i>	<i>v</i>
<i>Notations</i>	<i>vii</i>
<i>Introduction Générale</i>	<i>1</i>
1 <i>Optimisation combinatoire</i>	5
Introduction	5
1.1 C'est quoi l'optimisation combinatoire?	6
1.2 Modélisation des problèmes d'optimisation combinatoire	7
1.3 La complexité des problèmes d'optimisation combinatoire	11
1.4 Approches de résolution	14
2 <i>Approche polyédrale</i>	19
Introduction	19
2.1 La théorie polyédrale	20
2.2 Optimisation combinatoire et programmation linéaire	21
2.3 Séparation et optimisation	25
2.4 La technique de Lifting	26
3 <i>Le problème du sac à dos</i>	33
Introduction	33
3.1 Présentation du problème du sac à dos	34
3.2 Variantes et extensions du problème du sac à dos	39
3.3 Approches de résolution	44

3.4	Le polytope du problème de sac à dos	48
4	Lifting simultané de deux ensembles de variables	53
	Introduction	53
4.1	Lifting simultané de deux ensembles de variables	54
4.2	Comparaison	69
	Conclusion	71
5	Lifting simultané de plusieurs ensembles de variables	73
	Introduction	73
5.1	Lifting simultané de trois ensembles de variables	74
5.2	Lifting simultané de plusieurs ensembles de variables	84
	Conclusion	91
	Conclusion Générale	93
	Bibliographie	95

TABLE DES FIGURES

1.1	La liste des problèmes <i>NP</i> -Complets	12
2.1	Interprétation géométrique du lifting	31

LISTE DES TABLEAUX

3.1	La complexité des schémas d'approximation totalement polynomiaux (FPTAS) pour le problème du sac à dos	47
4.1	Les résultats du déroulement de l'exemple 4.1 pour $p = 2$ et $C_\Sigma = 57$	61
4.2	Les résultats du déroulement de l'exemple 4.1 pour $p = 1$ et $C_\Sigma = 28$	62
4.3	Les résultats du déroulement de l'exemple 4.1 pour $p = 0$ et $C_\Sigma = 0$	63
4.4	Les valeurs de $\alpha'_{p, E_\#^1}$ générées par l'algorithme 2	64
4.5	Les valeurs de $\alpha_{E_\#^1, E_\#^2}$ générées par l'algorithme 2	64
4.6	Les points affinement indépendants pour $E_\#^1 = 1$, $q_1 = 1$, $p = 2$ et $\alpha'_{2,1} = 1$	65
4.7	Les points affinement indépendants pour $E_\#^1 = 3$, $p = 2$, $q_1 = 2$ et $\alpha'_{2,3} = \frac{1}{2}$	66
4.8	Les points affinement indépendants pour $E_\#^1 = 4$, $E_\#^2 = 3$, $p = 3$, $q_1 = 3$, $q_2 = 2$ et $\alpha_{4,3} = \frac{1}{5}$	66
4.9	Résultats générés par MSLA	70
4.10	Les valeurs de $\alpha'_{p, E_\#^1}$ générées par l'algorithme 2	70
4.11	Les valeurs de $\alpha_{E_\#^1, E_\#^2}$ générées par l'algorithme 2	70
4.12	Les inégalités générées par MSLA et l'algorithme 2	71
5.1	Les résultats du déroulement de l'exemple 5.1, pour $p = 0$ et $C_\Sigma = 0$, avec $C = \{1, 2\}$, $E_1 = \{3, 4, 5, 6\}$, $E_2 = \{7, 8, 9\}$ et $E_3 = \{10, 11, 12\}$	78
5.2	Les valeurs de $\alpha'_{p, E_\#^1}$ générées par l'algorithme 3	79
5.3	Les valeurs de $\alpha_{E_\#^1, E_\#^2}$ générées par l'algorithme 3	79
5.4	Les valeurs de $\beta_{E_\#^2, E_\#^3}$ générées par l'algorithme 3	79

LISTE DES ALGORITHMES

1	Algorithme glouton pour le problème de sac à dos	48
2	Algorithme pour le lifting simultané de deux ensembles de variables	57
3	Algorithme pour le lifting simultané de trois ensembles de variables	77
4	Algorithme pour le lifting simultané de plusieurs ensembles de variables	87

Notations

\mathcal{A}	Algorithme d'approximation
C	Une couverture minimale $C = \{i_1, i_2, \dots, i_C\}$ pour \mathbb{P}
C_Σ	La somme de p éléments de la couverture minimale : $C_\Sigma = \sum_{r= C -p+1}^{ C } w_{i_r}$
d	La capacité d'un sac à dos
E	Un ensemble d'éléments fini
F	Face ou facette
\mathcal{F}	Une famille de sous ensembles de E
\mathbb{F}	Un élément de \mathcal{F}
I	Une instance d'un problème d'optimisation combinatoire
N	Un ensemble de n objet
\mathbb{P}	Le problème du sac à dos
\mathcal{P}	Le polytope du problème de sac à dos
P	Polyèdre ou polytope
p	Le nombre d'éléments pris de C correspondant aux plus petits coefficients w_j
S	Un sous ensemble de $\{0, 1\}^n$
X	Un ensemble convexe
C_p	Un sous ensemble de C ayant p éléments
E_i	Un sous ensemble de l'ensemble $N \setminus C : E_i = \{j_1^i, \dots, j_{ E_i }^i\}$
$E_{\#}^i$	Le nombre d'indices choisi de l'ensemble E_i pour être liftés
$E^{E_{\#}^i}$	Les $E_{\#}^i$ variables actuellement liftées
$E_{q_i}^{E_{\#}^i}$	Un sous ensemble de E_i ayant q_i éléments avec $E_{\#}^i$ variables liftées
E_{Σ}^i	La somme des q_i éléments de l'ensemble $E_i : E_{\Sigma}^i = \sum_{r=E_{\#}^i-q_i+1}^{E_{\#}^i} w_{j_r^i}$
p_j	Le profit de l'objet j
q_i	Le nombre maximum d'éléments qu'on nous pouvons prendre de l'ensemble $\{j_1^i, \dots, j_{E_{\#}^i}^i\}$
w_j	Le poids de l'objet j
$ I $	La taille de l'instance I en binaire
$ I_u $	La taille de l'instance I en unaire
Π	Problème d'optimisation combinatoire
λ_i	Le taux des éléments de E_i pris de $N \setminus C$, où, $\lambda_i \in]0, 1[$ pour $i = 1..m$
e	Un vecteur unitaire

Introduction générale

On qualifie généralement de *combinatoires* les problèmes dont la résolution se heurte à une explosion du nombre de combinaisons à explorer, tels que : le problème d'affectation, le problème d'arbre couvrant de poids minimal, ... Ces problèmes ont été étudiés depuis longtemps dans des axes de recherche indépendants et ce n'est qu'au milieu du vingtième siècle qu'ils ont été mis dans une seule structure en établissant des relations entre eux. Cette structure est nommée *optimisation combinatoire*.

Le problème du sac à dos est l'un des 21 problèmes NP-complets de Richard Karp, exposés dans son article [64] en 1972. Ce problème consiste à sélectionner un sous ensemble d'objets parmi n objets à mettre dans un sac de capacité d , où chaque objet possède un poids et un profit, de sorte que le poids total des objets sélectionnés n'excède pas la capacité d du sac tout en maximisant le profit total. C'est l'un des problèmes d'optimisation combinatoire le plus intensivement étudié depuis le milieu du XX^e siècle et on trouve des références dès 1897 [80]. La raison d'un tel intérêt est que le problème du sac à dos a de nombreuses applications et utilisations aussi bien en pratique que dans le domaine théorique.

Contrairement à l'optimisation classique (calcul différentiel, ...) où on a affaire à un domaine infini, l'optimisation combinatoire concerne le cas où le domaine est fini. Elle consiste à trouver un élément d'un ensemble fini qui maximise ou minimise une fonction donnée. Puisque l'ensemble des solutions est fini, on peut penser à énumérer toutes les solutions et à en choisir la meilleure. Malheureusement, ce raisonnement naïf atteint rapidement ses limites.

Vue la grande gamme de problèmes que l'on rencontre en pratique et qui peuvent être modélisés sous la forme d'un problème d'optimisation combinatoire, plusieurs approches ont été développées pour la résolution de ce type de problèmes et beaucoup d'améliorations ont été apportées sur le temps de leur résolution. L'une de ces approches, ayant connu un véritable essor dans les années 80, est l'approche polyédrale. Elle a été initiée par Edmonds [33] en 1965 pour le problème

du couplage et devient l'approche la plus utilisée pour la résolution des problèmes d'optimisation combinatoire. L'approche polyédrale consiste à ramener un problème d'optimisation combinatoire à un programme linéaire en décrivant l'enveloppe convexe de ses solutions par un système d'inégalités linéaires.

La caractérisation de l'enveloppe convexe des solutions par un système d'inégalités linéaires est l'étape la plus cruciale de l'approche polyédrale, car les problèmes d'optimisation combinatoire sont généralement NP-difficiles et par conséquent il y a peu d'espoir de trouver pour de tels problèmes une description complète explicite du polyèdre des solutions par un système d'inégalités linéaires. Pour ce, plusieurs méthodes de génération d'inégalités valides ont été développées, à savoir : la séparation, le renforcement, l'extension et la transformation d'inégalités valides, ...

La séparation d'inégalités valides est basée sur un problème dit de séparation lié au système des inégalités linéaires décrivant l'enveloppe convexe. En 1981, Grötschel et al [45] ont démontré que l'optimisation sur un polyèdre donné ne dépend pas du nombre de contraintes du système décrivant le polyèdre, mais plutôt du problème de séparation lié à ce système. Par conséquent, on peut résoudre un problème d'optimisation combinatoire en utilisant un algorithme séquentiel qui résout à chaque séquence un programme linéaire obtenu par l'ajout d'une contrainte violée, appelée coupe, déterminée par la résolution d'un problème de séparation. Un tel algorithme est dit de coupes ou de plans sécants.

Les coupes utilisées dans ce type d'algorithmes peuvent être partagées en deux classes : les coupes générales et les coupes portant sur des structures particulières. Les coupes générales peuvent être utilisées pour la résolution des programmes linéaires en nombres entiers ou mixtes, on cite : les coupes de Dantzig, les coupes de Chvátal-Gomory, les coupes mixtes de Gomory, les coupes lift-and-project; ... Les coupes portant sur des structures particulières proviennent de l'étude de problèmes combinatoires spécifiques telles que : les coupes de cliques, les coupes de cycle impair, les inégalités de couvertures et de couvertures étendues, ...

L'inégalité de couverture est l'une des plus importantes classes d'inégalités valides pour le problème du sac à dos. Ce type d'inégalité a été étudié par Balas [5], Hammer, Johnson et Peled [54], Wolsey [107] [108], et Balas et Zemel [6] lors de leur étude du polytope du sac à dos binaire. L'inégalité de couverture représente une face pour le polytope du problème de sac à dos et peut être liftée pour définir une facette.

Le lifting est un cas particulier d'extension d'inégalités valides. Cette technique est largement utilisée pour la génération des plans coupants forts. Depuis son introduction par Gomory en 1969 [41], elle a fait l'objet de plusieurs recherches [86] [109] [112] [6] et a été largement utilisée dans l'implémentation des algorithmes de plans coupants [23] [91]. Le lifting consiste à étendre une inégalité valide en y introduisant séquentiellement ou simultanément des variables qu'elle ne contient pas. D'où on peut distinguer deux types de lifting : le lifting séquentiel et le lifting simultané. Il y a aussi le lifting séquentiel indépendant qui utilise la notion de fonctions superadditives [51] [110].

Le lifting simultané a été introduit par Zemel en 1978 [112] pour des variables binaires. Depuis et durant deux décennies, personne n'a exploré ce sujet de recherche jusqu'en 2005 où Easton et Kevin [32] proposent un algorithme qui permet de lifter simultanément un ensemble de variables en une inégalité de couverture, pour le problème du sac à dos binaire, en un temps polynomial. En 2007, Sharma [98] a travaillé sur les perspectives de ces derniers auteurs et a développé un algorithme nommé MSLA (Maximal Simultaneously Lifted Algorithm) qui permet de trouver des inégalités de couverture maximales simultanément liftées pour un problème de sac à dos binaire.

Notre objectif est de développer des algorithmes qui permettent de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire.

Nous commençons d'abord par développer un algorithme qui permet de lifter simultanément deux ensembles de variables en une inégalité de couverture et nous énonçons ensuite les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du problème de sac à dos. Nous effectuerons après une comparaison, basée sur le déroulement d'un exemple, entre cet algorithme et MSLA. Ensuite, nous développons un autre algorithme qui permet de lifter simultanément trois ensembles de variables en une inégalité de couverture et nous énonçons également les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du problème de sac à dos. Nous terminons par une proposition de généralisation du lifting simultané sur plusieurs ensembles de variables en développant un autre algorithme qui permet de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture et nous énonçons les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du problème de sac à dos.

La thèse est structurée en cinq chapitres :

Dans *le premier chapitre*, nous présentons une jeune branche des mathématiques discrètes, nommée *optimisation combinatoire*. Nous commençons d'abord par définir c'est quoi un problème d'optimisation combinatoire. Puis, nous passons aux modèles généraux qui peuvent être utilisés pour la modélisation de ce type de problème. Par la suite, nous allons introduire quelques notions sur la théorie de la complexité. Et nous terminons par la présentation de quelques approches de résolution des problèmes d'optimisation combinatoire.

Le deuxième chapitre est consacré à l'une des approches de résolution des problèmes d'optimisation combinatoire, nommée *approche polyédrale*. Dans un premier temps, nous donnons quelques rappels sur la théorie polyédrale. Nous introduisons par la suite la relation entre l'optimisation combinatoire et : la programmation linéaire et la séparation. Finalement, nous allons présenter la technique du *lifting*.

Le problème du sac à dos fait l'objet du *troisième chapitre*. Nous commençons d'abord par la présentation du problème et l'énumération de quelques variantes. Nous introduisons par la suite les méthodes de résolution. Et nous terminons, par l'exploration du polytope du problème de sac à dos.

Notre *contribution* est présentée dans le *quatrième* et *cinquième chapitre* :

Dans le quatrième chapitre, nous commençons d'abord par présenter un algorithme qui permet de lifter simultanément deux ensembles de variables en une inégalité de couverture. Par la suite, nous allons effectuer une comparaison entre cet algorithme et MSLA.

Le cinquième chapitre fait l'objet de la généralisation du *lifting* simultané sur plusieurs ensembles de variables. Nous commençons d'abord par la présentation d'un algorithme qui permet de lifter simultanément trois ensembles de variables en une inégalité de couverture. Nous terminons par une proposition de généralisation du *lifting* simultané sur plusieurs ensembles de variables en présentant un autre algorithme qui permet de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture.

Optimisation combinatoire

L’optimisation combinatoire est l’une des plus jeunes branches des mathématiques discrètes. Dans le monde réel, on trouve beaucoup de problèmes qui peuvent être modélisés sous forme d’un problème d’optimisation combinatoire. Ce qui a incité plusieurs chercheurs à développer plusieurs approches pour la résolution de ce type de problèmes. Sous le terme optimisation combinatoire se regroupe un grand nombre de problèmes, à savoir : le problème du voyageur de commerce, le problème du sac à dos, le problème d’affectation, le problème de cheminement, le problème de couplage, le problème de flot, ...

La présentation de cette branche des mathématiques discrètes peut se faire de deux manières : par la présentation des problèmes ou par la présentation des approches de résolution. Dans ce chapitre, nous avons opté pour la présentation par les approches de résolution. Avant de commencer à les énumérer, nous allons d’abord définir c’est quoi un problème d’optimisation combinatoire. Puis, nous allons présenter quelques modèles généraux qui peuvent être utilisés pour la modélisation de ce type de problèmes. Ensuite, nous introduisons la théorie de la complexité.

1.1 C'est quoi l'optimisation combinatoire ?

En optimisation, nous sommes amenés à minimiser ou à maximiser une fonction donnée (soumise, ou non, à des contraintes) définie sur un certain domaine. Contrairement à l'optimisation classique (calcul différentiel, ...) où nous avons affaire à un domaine infini, l'optimisation combinatoire concerne le cas où le domaine est fini.

L'optimisation combinatoire peut être présentée soit par les problèmes qui y sont inclus ou par les approches de résolution. Nous optons pour la présentation par les approches de résolution, mais avant de commencer à les énumérer, nous allons d'abord définir formellement c'est quoi un problème d'optimisation combinatoire.

Définition 1.1. Soit :

- ◇ E un ensemble fini, $|E| = n$;
- ◇ $c = (c(e), e \in E)$ un vecteur poids associé aux éléments de E ;
- ◇ $\mathcal{F} \subseteq 2^E$ une famille de sous ensembles de E .
(Si $F \in \mathcal{F}$ alors $c(F) = \sum_{e \in F} c(e)$ est le poids de F).

Le problème d'optimisation combinatoire Π consiste à déterminer F^* dans \mathcal{F} tel que $c(F^*)$ soit maximum (minimum).

$$\Pi : \begin{cases} \max & c(F) = \sum_{e \in F} c(e) \\ & F \in \mathcal{F} \end{cases} \quad (1.1)$$

Nous pouvons dire que cela semble trivial à résoudre puisque que nous avons juste à choisir la meilleure solution parmi cet ensemble fini de solutions. Mais malheureusement, ce n'est pas le cas car les problèmes qualifiés d'être combinatoires se heurtent à une explosion du nombre de solutions à explorer.

Une instance I d'un problème d'optimisation combinatoire Π est une suite d'objets (des ensembles, des graphes, ...) et de nombres (des coûts, valeurs, tailles, ...). Formellement, nous pouvons définir une instance comme suit.

Définition 1.2. Une instance I d'un problème d'optimisation combinatoire Π peut être formulée par un couple (\mathcal{S}, g) , où :

- \mathcal{S} : est l'ensemble fini des solutions possibles ;

– g : est la fonction coût qui va de l'ensemble \mathfrak{S} vers l'ensemble des nombres réels \mathbb{R} :

$$g : \mathfrak{S} \longrightarrow \mathbb{R}$$

Définition 1.3. La taille $|I|$ d'une instance I est le nombre de bits utilisés pour écrire I en supposant que tous les nombres spécifiant l'instance sont écrits en binaire.

1.2 Modélisation des problèmes d'optimisation combinatoire

Un problème d'optimisation combinatoire peut être modélisé en utilisant différents formalismes. Nous allons nous restreindre ici aux modèles généraux, à savoir :

- ▷ La programmation linéaire ;
- ▷ La programmation linéaire en nombres entiers.

1.2.1 La programmation linéaire

Nous pouvons exprimer un problème d'optimisation combinatoire sous la forme d'un programme linéaire en variables continues (PL) qui s'écrit sous la forme générale suivante :

$$(PL) \left\{ \begin{array}{l} \max Z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = 1..m \\ x_j \geq 0 \quad j = 1..n \end{array} \right.$$

Les contraintes peuvent aussi être exprimées sous une forme matricielle :

$$(PL) \left\{ \begin{array}{l} \max Z = cx \\ Ax \leq b \\ x \geq 0 \end{array} \right.$$

où A représente la matrice des contraintes.

La forme standard d'un programme linéaire est :

$$(PL) \left\{ \begin{array}{l} \max Z = cx \\ Ax = b \\ x \geq 0 \end{array} \right.$$

Soit le vecteur $x = (x_1, \dots, x_n)^T$, où x est une solution de (PL) si et seulement si $Ax = b$. De plus, si $x \geq 0$, alors x est dite solution réalisable de (PL). La région admissible est l'ensemble D des solutions réalisables. Une solution optimale de (PL) est une solution réalisable qui maximise la valeur de la fonction objectif $Z = cx$.

La résolution graphique bien qu'elle permet de bien comprendre le rôle de la programmation linéaire mais elle est peu efficace car elle ne peut se faire que pour la dimension 2 et 3. Une méthode consiste à calculer la valeur de la fonction objectif pour tous les sommets admissibles à partir de la forme algébrique et à trouver la solution optimale du problème même si la dimension est supérieure à 3, mais le nombre de calcul associé explose rapidement.

L'algorithme du simplexe permet de diminuer fortement le nombre de calculs, car sa technique, qui a été mise au point par George Dantzig en 1947, a pour principe de passer d'une solution de base réalisable à une autre solution de base réalisable donnant la meilleure amélioration de la fonction objectif. Dans le cas où il n'est possible d'améliorer la valeur de la fonction objectif, la solution courante est optimale.

Géométriquement, l'algorithme du simplexe s'interprète comme un cheminement de point extrême en point extrême adjacent le long de la frontière de l'ensemble des solutions réalisables D .

Algébriquement, il s'interprète comme la détermination d'une suite de bases adjacentes B^0, B^1, \dots, B^q et de solutions de bases x^0, x^1, \dots, x^q , telles que :

$$z(x^0) \leq z(x^1) \leq \dots \leq z(x^q)$$

Bien que l'algorithme du simplexe résout les programmes linéaires en un temps exponentiel mais il est largement utilisé car il s'est révélé efficace en pratique et il constitue un outil de base très important pour l'optimisation combinatoire pour le fait que tout problème d'optimisation combinatoire que nous pouvons décrire par un système d'inégalités linéaires a un ensemble de solutions correspondant aux sommets du polyèdre délimité par ces inégalités.

Dans la théorie de la complexité, la programmation linéaire appartient à la classe P et cela est dû à l'algorithme de l'ellipsoïde introduit par Khachian en 1979.

Tout programme linéaire, appelé primal (PL), admet un programme linéaire dual (DL). Nous pouvons construire le dual (DL) d'un programme linéaire (PL) de la manière suivante :

- ◇ Le second membre des contraintes de (PL) représente les coefficients de la fonction objectif de (DL) ;
- ◇ Pour toute contrainte \geq du primal lui correspond une contrainte \leq dans le dual et vis versa ;
- ◇ Pour toute variable quelconque (sans restriction de signe : $\in \mathbb{R}$) du primal lui correspond une contrainte d'égalité dans le dual ;

- ◇ La matrice du dual est la transposée de la matrice du primal ;
- ◇ Si le problème primal (PL) est un problème de maximisation alors le problème dual (DL) devient un problème de minimisation et vis versa.

$$(PL) \begin{cases} \max & Z = c^T x \\ & Ax \leq b \\ & x \geq 0 \end{cases} \qquad (DL) \begin{cases} \min & W = b^T y \\ & A^T y \geq c \\ & y \geq 0 \end{cases}$$

Les propriétés et les structures du primal et du dual sont symétriques. Connaissant la solution optimale de l'un d'entre eux nous pouvons déduire la solution optimale de l'autre. Le théorème suivant, appelé théorème des écarts complémentaires, permet de caractériser la condition d'optimalité du primal et du dual.

Théorème 1.1. *Soit x^* et y^* des solutions admissibles respectivement du primal et du dual. Une condition nécessaire et suffisante pour que x^* et y^* soient des solutions optimales est qu'elles vérifient les relations de complémentarité :*

$$y^{T*} (Ax^* - b) = 0 \tag{1.2}$$

$$x^{T*} (A^T y^* - c) = 0 \tag{1.3}$$

1.2.2 La programmation linéaire en nombres entiers

Depuis les premiers travaux de Gomory [40], la programmation linéaire en nombres entiers représente le domaine de la programmation mathématique le plus riche et le plus actif. Le volume des publications et des recherches qui lui ont été consacrées atteste la difficulté et l'importance des applications.

Nous parlons de la programmation linéaire en nombres entiers lorsque le domaine des solutions d'un programme linéaire est restreint à des valeurs entières, car c'est le cas que nous rencontrons dans de nombreuses situations réelles.

Selon les valeurs prises par les variables du programme linéaire en nombres entiers, nous pouvons distinguer :

- Les programmes linéaires en variables entières : Lorsque toutes les variables du problème sont entières ;
- Les programmes linéaires en variables mixtes : Nous trouvons des variables entières et des variables continues ;

- Les programmes linéaires en variables binaires (ou en 0 – 1) : Les variables prennent leurs valeurs dans l'ensemble $\{0, 1\}$.

La forme générale d'un programme linéaire en nombres entiers (*PLNE*) est la suivante :

$$(PLNE) \begin{cases} \max Z = & cx \\ & Ax \leq b \\ & x \in \mathbb{N} \end{cases}$$

Le problème (*PL*) obtenu à partir de (*PLNE*) en relâchant (i.e. en oubliant) les contraintes d'intégrité sera appelé relaxation continue de (*PLNE*), ou encore, programme linéaire (continu) associé au programme linéaire en nombres entiers.

Contrairement aux programmes en variables continues, la résolution des programmes linéaires en nombres entiers n'est généralement pas aisée, car ils appartiennent à la classe des problèmes *NP*-difficiles.

Les deux principales familles de méthodes actuellement connues pour la résolution des programmes linéaires en nombres entiers sont : les approches par énumération implicite et les méthodes de coupes.

Les méthodes de coupes ont permis de résoudre des problèmes particuliers de type partitionnement ou recouvrement (Delorme 1975 [29]). Leur principe consiste à ajouter des contraintes linéaires une par une, appelées coupes, n'excluant aucun point entier admissible jusqu'à ce que la solution optimale de la relaxation soit entière.

Dans la littérature, nous trouvons plusieurs types de coupes [75]. Celles-ci peuvent être partitionnées en deux classes :

- **Les coupes générales** : Ces coupes n'ont aucune structure particulière requise. Elles peuvent être utilisées pour la résolution des programmes linéaires en nombres entiers ou mixtes :
 - ▷ Coupes de Dantzig ;
 - ▷ Les coupes de Chvátal-Gomory ;
 - ▷ Coupes mixtes de Gomory ;
 - ▷ Les coupes lift-and-project ;
 - ▷ ...

- **Les coupes portant sur des structures particulières** : Si nous limitons notre attention à une seule contrainte, ou à un sous ensemble de contraintes, même un problème général peut avoir des structures locales, par exemple : toutes les variables qui apparaissent dans les contraintes sont binaires, ou une partie du problème linéaire peut être un problème de flot, ... Nous citons à titre d'exemple :
 - ▷ Coupes de clique ;
 - ▷ Coupes de cycle impair ;
 - ▷ Inégalités de couverture ;
 - ▷ ...

1.3 La complexité des problèmes d'optimisation combinatoire

Les problèmes d'optimisation combinatoire sont formellement caractérisés par la théorie de la complexité, qui propose une classification des problèmes en fonction de la complexité de leur résolution.

1.3.1 Complexité théorique

La complexité d'un problème est le nombre d'instructions estimées pour sa résolution par rapport à la taille de l'instance la plus difficile (pire des cas).

Les classes de complexité ont été introduites pour les problèmes de décision dont la réponse est *oui* ou *non*.

Définition 1.4. [33] Un problème est dit polynomial, s'il existe un algorithme pour lequel le temps maximum nécessaire à la résolution de chaque instance, de ce problème, est borné par une fonction polynomiale de la taille de cette instance.

Définition 1.5. *Un algorithme non déterministe* est un algorithme qui, en plus des instructions usuelles, comporte des instructions de choix.

1.3.1.1 La classe P et la classe NP

Pour les problèmes de décision, nous distinguons deux classes : la classe P et la classe NP . La classe P regroupe les problèmes polynomiaux. Mais malheureusement, tous les problèmes combinatoires ne sont pas polynomiaux comme l'a montré Meyer Stockmeyer en 1973 [101] pour certains problèmes de mots. La classe NP est la classe des problèmes qui peuvent être résolus en un temps polynomial par un algorithme non déterministe.

Ces deux classes sont caractérisées par une relation d'inclusion qui se résume par la célèbre conjecture :

$$P \neq NP$$

1.3.1.2 La classe des problèmes NP-complets

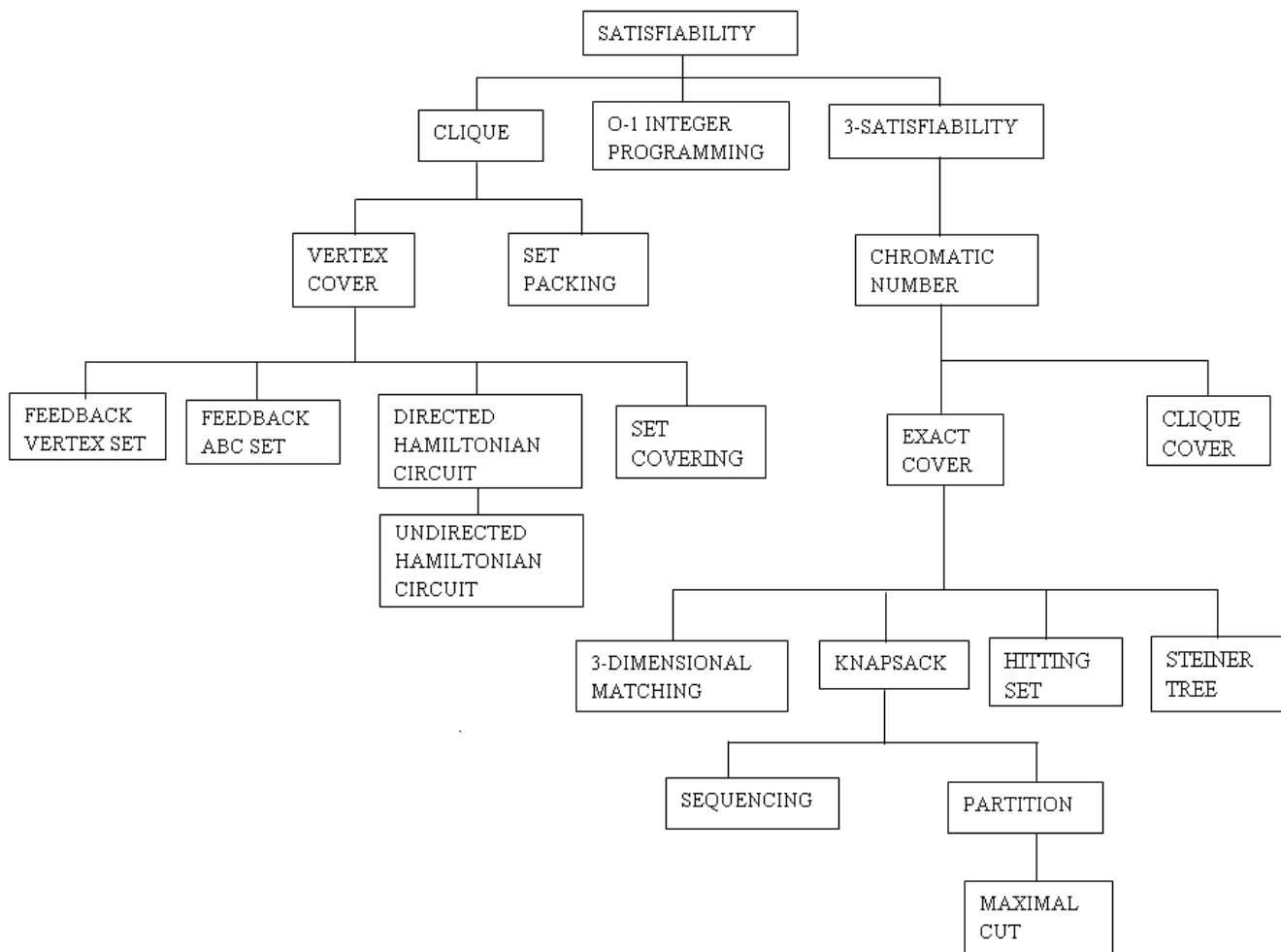


FIG. 1.1: La liste des problèmes NP-Complets

1.3.1.3 \leftrightarrow Les problèmes NP-difficiles

Définition 1.6. Un problème est dit NP-difficile, si tout problème de NP lui est faiblement réductible.

1.3.1.4 \hookrightarrow Les problèmes *NP-complets*

Définition 1.7. Un problème est dit *NP-complet*, s'il est *NP*-difficile et *NP*.

Considérons le problème de réalisation d'une expression logique :

Théorème 1.2 (Cook (1971)). [21] *Le problème de réalisation d'une expression logique est NP-complet.*

Les problèmes *NP-complets* sont donc les problèmes équivalents faiblement au problème de la réalisation d'une expression logique.

1.3.2 Les problèmes d'optimisation combinatoire et la complexité

A chaque problème d'optimisation combinatoire, nous pouvons associer un problème de décision dont le but est de déterminer s'il existe une solution pour laquelle la fonction objectif soit supérieure (resp. inférieure) ou égale à une valeur donnée. La complexité d'un problème d'optimisation combinatoire est liée à celle du problème de décision qui lui est associé. En particulier, si le problème de décision est *NP-complet*, alors le problème d'optimisation combinatoire est dit *NP-difficile*.

1.3.3 En pratique

Théoriquement, il est illusoire de chercher un algorithme polynomial pour un problème *NP-difficile* (à moins que $P = NP$). En pratique, ce n'est pas le cas car il apparaît que certaines instances sont beaucoup plus faciles que d'autres et peuvent être résolues très rapidement. Ceci est dû au fait que la théorie de la complexité se base sur l'instance la plus difficile (pire des cas) d'un problème donné pour évaluer sa complexité.

Dans la pratique [100], les types d'instances que nous pouvons rencontrer sont :

- **Les instances ayant très peu de contraintes d'exclusion** : Elles sont faciles à résoudre, car elles admettent beaucoup de solutions ;
- **Les instances ayant beaucoup de contraintes** : Ces instances sont aussi facilement traitables, car nous arrivons rapidement à montrer qu'elles n'admettent pas de solutions ;
- **Les instances intermédiaires** : Ces instances sont généralement beaucoup plus difficiles à résoudre, car elles ont trop de contraintes pour trouver une solution, mais pas assez pour montrer qu'elles n'ont pas de solutions, facilement.

1.4 Approches de résolution

1.4.1 Approches par énumération implicite

Pour résoudre un problème d'optimisation combinatoire nous pouvons penser à explorer toutes les 2^n solutions possibles et à en choisir la meilleure. Mais malheureusement, cette méthode naive atteint rapidement ces limites. L'approche par énumération implicite consiste en une énumération intelligente du domaine des solutions, en se basant sur le principe *diviser pour régner*. C'est une approche exacte qui s'applique lorsqu'il est possible de diviser un problème en sous problèmes indépendants.

L'utilisation de cette approche remonte aux travaux de Land et Doig (1960) [69] et Dakin (1965) [25]. Elle a été introduite par Little et al [71], en 1963, pour le problème du voyageur en commerce, puis elle a été reprise par plusieurs auteurs sous différentes variantes.

L'approche par énumération implicite est connue sous d'autres noms, à savoir : *SEP* ou *PSEP* (pour Procédure par Séparation et Evaluation Progressive), *SES* ou *PSES* (pour Procédure par Séparation et Evaluation Séquentielle), algorithme A^* en intelligence artificielle, Branch and Bound en anglais, énumération arborescente par séparation et élagage ou bien méthode arborescente par séparation et évaluation.

Les approches par énumération implicite ont en commun, l'élaboration d'une arborescence de recherche dont le développement depuis la racine se fait à l'aide de trois ingrédients essentiels :

- **Le principe de séparation :** La mise en œuvre des approches par énumération implicite est conditionnée par le principe de séparation car c'est lui qui définit la façon de construire les sommets de l'arborescence ;
- **Les principes d'élagage :** Nous utilisons en général des fonctions d'évaluation et des bornes ;
- **La stratégie de développement de l'arborescence :** L'exploration des sommets de l'arborescence des solutions obéit à une stratégie donnée. Dans la littérature, nous distinguons trois stratégies différentes :
 - ✓ Profondeur d'abord ;
 - ✓ Largeur d'abord ;
 - ✓ Meilleur d'abord.

Au cours du processus de développement de l'arborescence, l'ensemble des solutions est partitionné en deux ou plusieurs sous ensembles en éliminant les parties ne contenant pas de solutions entières réalisables. Chaque sommet représente un sous problème et la racine correspond au problème tout entier. A chaque itération, nous sélectionnons, en utilisant une stratégie de sélection, un sous ensemble promoteur conduisant à une meilleure solution réalisable. Si cette solution est trouvée, nous éliminons le sous ensemble des prochaines considérations et nous obtenons ainsi une borne inférieure pour le problème en question, sinon nous continuons l'exploration.

Dans certains cas, l'approche par énumération implicite n'aboutit pas à une solution optimale entière. Mais cela n'empêche pas de l'utiliser pour la résolution des programmes linéaires en nombres entiers, et ce en la combinant avec d'autres techniques, à savoir : les méthodes de coupes et la génération de colonnes. Ces combinaisons ont donné lieu à de nouvelles approches de résolution très efficaces, nous citons :

- ◇ **Branch and Cut** : Cette approche a été introduite par Padberg et Rinaldi(1991) [85] dans le contexte de la résolution du problème du voyageur en commerce. Branch and cut vient de l'idée de rajouter des coupes à chaque nœud de l'arborescence de recherche. Si l'ajout des plans coupants se restreint à la racine, nous utilisons le terme *Cut and Branch*.
- ◇ **Branch and Price** : [10] Vient de l'idée de combiner la génération de colonnes avec Branch and Bound.

1.4.2 Approches approximatives

La plupart des problèmes d'optimisation combinatoire sont *NP*-difficiles. Pour ce type de problèmes, les méthodes exactes impliquent un temps de calcul énorme ou atteignent rapidement leurs limites sans qu'une solution optimale soit trouvée. Dans ce cas, nous pouvons nous contenter d'une solution approximative avec une erreur permise bornée par une constante ϵ . En effet, beaucoup d'algorithmes d'approximation ont été développés pour les problèmes d'optimisation combinatoire [104].

Un algorithme d'approximation \mathcal{A} pour un problème d'optimisation combinatoire Π calcule en un temps polynomial (en la taille de l'instance) une solution réalisable dont la valeur de la fonction objectif est *proche* de l'optimum. La définition formelle est différente selon si le problème est de minimisation ou de maximisation. Considérons un problème d'optimisation combinatoire *NP*-difficile Π , de fonction objectif f_{Π} .

Définition 1.8. [104] Étant donné un problème de minimisation (resp : maximisation) Π , et une fonction :

$$\delta : \mathbb{N} \longrightarrow \mathbb{Q}^+$$

telle que $\delta \geq 1$ (resp : $\delta \leq 1$), un algorithme \mathcal{A} est une δ -approximation pour Π si pour toute instance I , \mathcal{A} produit en temps polynomial en $|I|$, une solution réalisable s de I telle que :

$$f_{\Pi}(I, s) \leq \delta(|I|) \cdot OPT(I) \quad (\text{resp} : f_{\Pi}(I, s) \geq \delta(|I|) \cdot OPT(I)).$$

Clairement, plus le facteur d'approximation δ est proche de 1, l'algorithme d'approximation est meilleur.

1.4.2.1 Les problèmes d'optimisation combinatoire *NP*-difficiles au sens faible

Les problèmes d'optimisation combinatoire *NP*-difficiles au sens faible présentent la particularité de pouvoir être approchés par des schémas d'approximation totalement polynomiaux (*FPTAS*). Ce qui n'est pas le cas pour les problèmes d'optimisation combinatoire *NP*-difficiles au sens fort, car sinon $P = NP$. Ceci est justifié par les résultats suivants.

Théorème 1.3. Soient p un polynôme et Π un problème *NP*-difficile de minimisation tel que sa fonction objectif f_{Π} ne prenne que des valeurs entières et tel que pour toute instance I , $OPT(I) < p(|I_u|)$. Si Π admet un *FPTAS*, alors il admet aussi un algorithme pseudo-polynomial.

Corollaire 1.4. Soit Π un problème d'optimisation *NP*-difficile vérifiant les hypothèses du théorème 1.3. Si Π est *NP*-difficile au sens fort, alors Π n'admet pas de *FPTAS*, à moins que $P = NP$.

Un algorithme \mathcal{A} est un schéma d'approximation (Approximation scheme) pour Π si pour toute entrée (I, ϵ) , où I est une instance de Π et $\epsilon > 0$ est la précision demandée, la solution trouvée s vérifie :

- $f_{\Pi}(I, s) \leq (1 + \epsilon) \cdot OPT$, si Π est un problème de minimisation.
- $f_{\Pi}(I, s) \geq (1 + \epsilon) \cdot OPT$, si Π est un problème de maximisation.

\mathcal{A} est un schéma d'approximation polynomial (Polynomial time approximation scheme : *PTAS* en abrégé), si pour tout $\epsilon > 0$ fixé, le temps de calcul de \mathcal{A} est borné par un polynôme en la taille de l'instance I . Si \mathcal{A} est un schéma d'approximation polynomial (*PTAS*) dont le temps de calcul est borné par un polynôme en la taille de l'instance I et en $\frac{1}{\epsilon}$, pour tout (I, ϵ) alors \mathcal{A} est appelé un schéma d'approximation totalement polynomial (Fully polynomial time approximation scheme : *FPTAS* en abrégé). Techniquement, un *FPTAS* est ce que nous pouvons espérer de mieux pour un problème d'optimisation combinatoire *NP*-difficile, à moins que $P = NP$.

1.4.3 Approches heuristiques

Les heuristiques sont des méthodes qui utilisent une recherche guidée par des *astuces*, qui dépendent du problème traité, pour approcher une solution optimale (sous-optimale), nous les appelons parfois méthodes approchées. Elles peuvent être conçues pour résoudre un type de problèmes donnés. En général, elles ne prennent pas un nombre exponentiel d'itérations et nous pouvons observer empiriquement qu'elles trouvent une bonne solution rapidement, ce qui les rend utiles pour les problèmes nécessitant une solution en temps réel même s'ils sont *NP*-difficiles. Elles peuvent aussi être utilisées afin d'initialiser une méthode exacte (Branch & Bound par exemple). Leur inconvénient est qu'elles ne produisent pas de mesure de qualité de la solution.

Une métaheuristique est une méthode générale (recuit simulé, tabou search, ...), qui peut être adaptée à divers problèmes d'optimisation.

Les approches heuristiques peuvent être classées en différentes catégories [37] :

- ◇ **Constructives** : Algorithmes gloutons, méthode Pilote, GRASP ;
- ◇ **Recherche locale** : Algorithmes de descente, multi-départs, recuit simulé, algorithme à seuil, recherche Tabou, méthode de bruitage ;
- ◇ **Évolutionnistes** : Algorithmes génétiques, algorithmes d'évolution, recherche dispersée, méthode des chemins, colonie de fourmis ;
- ◇ **Réseaux de neurones** : Modèle de Hopfield-Tank, machine de Boltzmann, réseau auto-adaptatif, réseau élastique ;
- ◇ **Heuristiques Bayésiennes** : Optimisation globale, optimisation discrète ;
- ◇ **Superposition** : Perturbation des données, perturbation des paramètres d'une heuristique.

1.4.4 Approches polyédrales

Les méthodes de coupes présentent l'inconvénient de ne pas générer les coupes les plus profondes, elles peuvent aussi conduire à engendrer un nombre extrêmement élevé de coupes. Dans les années 80, une autre approche, élaguant les inconvénients des méthodes de coupes, est apparue. Cette approche est connue sous le nom *combinatoire polyédrique* ou *approche polyédrale*.

Approche polyédrale

L'approche polyédrale est l'une des approches les plus utilisées pour la résolution des problèmes d'optimisation combinatoire. Celle-ci consiste à ramener un problème d'optimisation combinatoire à un programme linéaire en décrivant l'enveloppe convexe de ses solutions par un système d'inégalités linéaires. Pour ce, plusieurs méthodes de génération d'inégalités valides ont été développées [13], à savoir : la séparation, le renforcement, l'extension et la transformation d'inégalités valides. La technique de *lifting* est un cas spécial d'extension d'une inégalité valide, elle consiste à prendre une inégalité valide pour un polyèdre donné et la rendre valide pour un espace de dimension supérieure.

Dans ce chapitre, nous allons introduire l'approche polyédrale. Nous commençons d'abord par quelques rappels sur la théorie polyédrale. Ensuite, nous allons décrire la nature de la relation entre l'optimisation combinatoire et : la programmation linéaire et la séparation. Et nous terminons par la présentation de la technique de *lifting* qui fait l'objet de notre étude.

2.1 La théorie polyédrale

L'analyse convexe est une branche des mathématiques dans laquelle nous étudions les ensembles convexes. Un ensemble convexe X est un ensemble sans trou i.e. $\forall x, y \in X, \forall \lambda \in [0, 1]$ le point $z = (\lambda x + (1 - \lambda)y) \in X$. Géométriquement, cela veut dire qu'un ensemble convexe contient un segment entre chaque pair de ses points.

La théorie polyédrale est une partie de l'analyse convexe qui s'intéresse à une classe importante d'ensembles convexes, appelée *polyèdres*. L'importance de cette classe réside dans le fait qu'elle représente les ensembles des solutions faisables des programmes linéaires. En effet, un polyèdre est l'intersection d'un nombre fini de demi-espaces. Ces derniers peuvent représenter les espaces de solutions délimités par les inégalités linéaires d'un programme linéaire car par définition un demi-espace est l'espace des solutions d'une inégalité linéaire, i.e. $\forall x \in \mathbb{R}^n$ tel que : $\sum_{i=1}^n \pi_i x_i \leq \pi$. Un polyèdre borné est appelé polytope. Un cône est un polyèdre K défini par : $K = \{r \in \mathbb{R}^n : Ar \leq 0\}$. Le cône polaire $K^0 \subseteq \mathbb{R}^n$ associé à un cône convexe $K \subseteq \mathbb{R}^n$ est $K^0 = \{y \in \mathbb{R}^n : y^T x \leq 0 \quad \forall x \in K\}$.

Dans l'optimisation combinatoire, un problème peut toujours être formulé sous la forme d'un programme linéaire en nombres entiers. Dans un premier temps, nous ne voyons pas le lien entre l'optimisation combinatoire et la théorie polyédrale car l'ensemble des solutions d'un programme linéaire en nombres entiers X , qui est discret (ensemble de points), n'est ni convexe ni un polyèdre? Néanmoins, nous pouvons définir pour l'ensemble X une enveloppe convexe $conv(X)$ qui est le plus petit ensemble convexe contenant X . Pour ce faire, nous délimitons les points de X en utilisant des hyperplans H définis par les inégalités linéaires du programme linéaire relaxé, où $H = \{x \in \mathbb{R}^n : \pi^T x = \pi_0, \pi \in \mathbb{R}^n, \pi_0 \in \mathbb{R}\}$. Il est clair que $conv(X)$ est un polyèdre.

Une inégalité linéaire $\pi x \leq \pi_0$ est dite valide pour un polyèdre $P \subseteq \mathbb{R}^n$ si elle est vérifiée pour tout point de P , i.e., $P \subseteq \{x \in \mathbb{R}^n \mid \pi x \leq \pi_0\}$. Si de plus, il existe $x \in P$ tel que $\pi x = \pi_0$, elle est dite d'appui. Le polyèdre $F = \{x \in P \mid \pi x = \pi_0\}$ est appelé face de P , nous dirons que F est défini par l'inégalité $\pi x \leq \pi_0$. Si cette dernière n'est pas d'appui, alors F est une face vide, et si $F \neq \emptyset$ et $F \neq P$, alors F est dite face propre de P . Une face de P de dimension 0 est appelée point extrême.

Nous dirons qu'une inégalité $\pi x \leq \pi_0$, valide pour P , induisant une face F_1 de P , domine une inégalité $\pi' x \leq \pi'_0$, également valide pour P , et induisant une face F_2 de P , si F_2 est strictement incluse dans F_1 . Lorsque les deux inégalités $\pi x \leq \pi_0$ et $\pi' x \leq \pi'_0$ apparaissent dans la description linéaire du polyèdre P , nous pouvons omettre l'inégalité $\pi' x \leq \pi'_0$ dans la description de P . L'in-

égalité $\pi' x \leq \pi'_0$ est dite redondante. En revanche, si nous avons $F_1 = F_2$, alors les deux inégalités sont dites équivalentes.

La dimension d'un polyèdre est le nombre maximum de ses points affinement indépendants moins 1. Par convention, la dimension d'un ensemble vide est -1 . Les points x_1, \dots, x_n sont dits affinement indépendants si nous avons $\sum_{i=1}^k \lambda_i x^i = 0$ et $\sum_{i=1}^k \lambda_i = 0$, alors $\lambda_i = 0, \forall i \in \{1, \dots, k\}$ est l'unique solution. La dimension d'un polyèdre joue un rôle important dans la détermination des inégalités valides nécessaires à la description d'un polyèdre. En effet, si une inégalité valide induit une face F de dimension exactement inférieure de 1 de la dimension de P , i.e. $\dim(F) = \dim(P) - 1$, alors cette inégalité, nommée facette, est nécessaire dans la description de P .

Le théorème suivant est communément utilisé pour démontrer si une inégalité valide donnée représente une facette :

Théorème 2.1. [84] Soit $\alpha^T x \leq \beta$ une inégalité valide, s'il existe un point $x \in \text{conv}(P)$ tel que $\alpha^T x < \beta$, alors $\dim(x \in \text{conv}(P) : \alpha^T x = \beta) \leq \dim(\text{conv}(P)) - 1$.

Nous pouvons montrer qu'une inégalité valide donnée représente une facette F de P de différentes manières [73] :

- ◇ **Preuve de nécessité :** En utilisant l'une des propriétés de facettes, à savoir la propriété de nécessité, nous pouvons montrer qu'une inégalité donnée : $\pi x \leq \pi_0$, représente une facette F de P en montrant que F est nécessaire dans la description de P .
- ◇ **Preuve directe :** La dimension d'une facette F de P est $\dim(F) = \dim(P) - 1$, d'où la technique la plus directe pour montrer qu'une inégalité valide $\pi x \leq \pi_0$ induit une facette F de P consiste à exhiber $\dim(P)$ solutions affinement indépendantes et satisfaisant l'inégalité à égalité.
- ◇ **Preuve par maximalité :** Une dernière méthode pour montrer qu'une face F induite par $\pi x \leq \pi_0$ est une facette consiste à montrer qu'elle n'est pas strictement contenue dans une facette de P .

2.2 Optimisation combinatoire et programmation linéaire

L'optimisation combinatoire consiste à optimiser une fonction donnée dans un ensemble fini de solutions. Ce qui n'est pas le cas pour la programmation linéaire, où l'optimisation se fait dans un ensemble convexe qui est infini. Quel est donc le lien entre l'optimisation combinatoire et la

programmation linéaire? Et quel est l'intérêt de passer de l'optimisation dans un ensemble fini à l'optimisation dans un ensemble infini? Dans cette section, nous allons essayer de répondre progressivement à ces deux questions.

2.2.0.1 \hookrightarrow Formulation d'un problème d'optimisation combinatoire sous la forme d'un programme linéaire en nombres entiers

Soient E un ensemble fini et \mathcal{F} une famille de sous ensembles de E . Pour $F \in \mathcal{F}$, nous associons un vecteur $x^F \in \{0, 1\}^E$, appelé vecteur d'incidence donné par :

$$x_i^F = \begin{cases} 1 & \text{si } i \in F; \\ 0 & \text{si } i \in E \setminus F. \end{cases} \quad (2.1)$$

Soit :

$$\mathcal{S} = \{x^F : F \in \mathcal{F}\} \quad (2.2)$$

l'ensemble des vecteurs d'incidence des solutions de (1.1), alors le problème (1.1) est équivalent au problème suivant :

$$\max\{cx : x \in \mathcal{S}\} \quad (2.3)$$

qui peut être formulé comme un programme linéaire en nombres entiers de la forme :

$$\begin{cases} \max & Z = cx \\ & Ax \leq b \\ & 0 \leq x \leq 1 \\ & x \text{ entier} \end{cases}$$

Par la proposition suivante, nous avons la valeur optimale du problème (2.3) est égale à celle du problème suivant :

$$\max\{cx : x \in \text{conv}(\mathcal{S})\} \quad (2.4)$$

Proposition 2.2. Soient $X \subseteq \mathbb{R}^n$ un ensemble de points et c un vecteur de \mathbb{R}^n , alors :

$$\max\{cx : x \in X\} = \max\{cx : x \in \text{conv}(X)\} \quad (2.5)$$

2.2.0.2 \hookrightarrow Equivalence entre un problème d'optimisation combinatoire et le programme linéaire en nombres entiers qui lui est associé

Le théorème suivant montre que le problème d'optimisation combinatoire et le programme linéaire en nombres entiers qui lui est associé sont équivalents.

Théorème 2.3. Un ensemble (non vide) de points $P \subseteq \mathbb{R}^n$ est un polytope si et seulement s'il existe un ensemble de points S tel que $P = \text{conv}(S)$.

D'après le théorème 2.3, $\text{conv}(\mathcal{S})$ est un polytope, et peut donc être décrit par un système fini d'inégalités linéaires. Par conséquent, le problème (2.4) n'est rien d'autre qu'un programme linéaire.

Théorème 2.4 (Dantzig 1947). *Une solution optimale d'un programme linéaire $\max\{cx \mid Ax \leq b\}$ peut être prise parmi les points extrêmes du polyèdre défini par ses contraintes.*

Théorème 2.5. *Pour tout point extrême x^* de P , il existe $c \in \mathbb{Z}^n$ tel que x^* est la solution optimale unique du programme $\max\{cx : x \in P\}$*

Théorème 2.6. *Supposons que $\text{rang}(A) = n$, et le problème*

$$\max\{cx : x \in P\} \tag{2.6}$$

admet une solution optimale finie. Alors il existe une solution optimale de (2.6) qui est un point extrême de P .

Comme les points extrêmes de $\text{conv}(\mathcal{S})$ sont les solutions de \mathcal{S} , il en résulte que les problèmes (2.3) et (2.4) sont équivalents.

2.2.0.3 \hookrightarrow Approche polyédrale

L'équivalence entre un problème d'optimisation combinatoire et le programme linéaire en nombres entiers qui lui est associé a donné naissance à une nouvelle approche appelée *approche polyédrale*. Cette approche est initiée par Jack Edmonds (1965) [33], pour le problème de couplage. Elle consiste à ramener un problème d'optimisation combinatoire à un programme linéaire en décrivant l'enveloppe convexe de ses solutions, appelée le polytope associé au problème (ou le polytope des solutions du problème), par un système d'inégalités linéaires. Les étapes de l'approche polyédrale peuvent se résumer comme suit :

1. Représenter les éléments de \mathcal{F} par un ensemble de vecteurs en 0 – 1 ;
2. Définir le polytope P , enveloppe convexe des points de \mathcal{S} ;
3. Déterminer un système complet qui décrit P ;
4. Appliquer la programmation linéaire pour résoudre le problème.

L'étape cruciale de l'approche polyédrale est l'étape 3. La difficulté de cette étape se résume dans les trois points suivants :

1. Nous avons des résultats théoriques qui impliquent l'existence d'un système linéaire fini qui décrit P , mais n'indiquent pas comment l'obtenir ;
2. Si le problème en question est NP -complet, il y a peu d'espoir d'obtenir une description complète de P ;
3. Si nous arrivons à surmonter les deux problèmes précédents et trouver une description complète du polyèdre P , celle ci peut être de grande de taille (exponentielle). Cependant nous ne pouvons pas l'exploiter entièrement pour résoudre le problème comme un programme linéaire.

Mais quel est donc le secret de cette approche à être largement utiliser malgré toutes les difficultés qu'elle présente ?

2.2.0.4 \hookrightarrow Dominant d'un polyèdre

Définition 2.1. Le dominant d'un polyèdre $P \subseteq \mathbb{R}^n$, noté par $dom(P)$, est le polyèdre qui consiste en les points x tels que $x \geq x'$ pour certain $x' \in P$, i.e.

$$dom(P) = P + \mathbb{R}_+^n \quad (2.7)$$

Il est clair que $P \subseteq dom(P)$ et $dom(P)$ est non borné. Le dominant d'un polyèdre est généralement plus simple à caractériser. De plus, en utilisant la propriété algorithmique suivante, nous pouvons considérer le dominant de P au lieu d'étudier P :

Remarque 2.1. Si $c \in \mathbb{R}_+^n$, alors :

$$\min\{cx \mid x \in P\} = \min\{cx \mid x \in dom(P)\} \quad (2.8)$$

2.2.0.5 \hookrightarrow Dualité et optimisation combinatoire :

La notion de la dualité dans la programmation linéaire peut avoir des applications intéressantes en optimisation combinatoire. En effet, si nous avons deux problèmes d'optimisation combinatoire Π_1 et Π_2 dont les systèmes décrivant leurs polyèdres des solutions sont duaux, à savoir :

$$\Pi_1 : \quad \max\{cx : Ax \leq b, x \geq 0\} \quad (2.9)$$

$$\Pi_2 : \quad \min\{b^T y : A^T y \leq c^T, y \geq 0\} \quad (2.10)$$

alors nous pouvons obtenir une relation min-max entre les solutions de ces deux problèmes :

$$\max\{c(F) : F \in \mathcal{F}_1\} = \min\{b(F) : F \in \mathcal{F}_2\} \quad (2.11)$$

La notion de la dualité en optimisation combinatoire a permis d'introduire la classe des matrices totalement unimodulaires et les systèmes totalement duaux entiers :

- **La classe des matrices totalement unimodulaires :** Une matrice $A_{m \times n}$ est dite totalement unimodulaire (*TU*) si le déterminant de toute sous-matrice carrée de A est 0, 1 ou -1 . Il est à noter que la transposée d'une matrice totalement unimodulaire est totalement unimodulaire. Le lien fondamental entre cette classe de matrices et les polyèdres est établi par le théorème suivant.

Théorème 2.7 (Hoffman, Kruskal (1956)). [57] Une matrice $A_{m \times n}$ est totalement unimodulaire si et seulement si pour tout vecteur entier $b \in \mathbb{R}^m$, le polyèdre $\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ est entier.

- **Les systèmes totalement duaux entiers :** Le système linéaire $Ax \leq b$ est dit totalement dual entier (*TDE*) si pour tout vecteur entier $c \in \mathbb{R}^n$, tel que le programme linéaire $\max\{cx : Ax \leq b\}$ admet une solution optimale, alors le programme dual correspondant possède une solution optimale entière. Avec $A_{m \times n}$ une matrice rationnelle et $b \in \mathbb{R}^m$. Le théorème suivant nous donne une condition suffisante pour qu'un polyèdre soit entier.

Théorème 2.8 (Edmonds, Giles (1977)). [34] Si $Ax \leq b$ est *TDE* et b est entier, alors le polyèdre $\{x \in \mathbb{R}^n : Ax \leq b\}$ est entier.

2.3 Séparation et optimisation

Pour les problèmes d'optimisation combinatoire de la classe *NP* il y a peu d'espoir de trouver une description linéaire complète du polyèdre des solutions. Néanmoins, nous avons vu dans la section précédente que même en n'ayant pas une description linéaire complète, nous pouvons savoir si le polyèdre $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ décrivant un problème donné est entier ou non, en vérifiant si le système $Ax \leq b$ est *TDE* ou si la matrice A est *TU*. Malheureusement, ce n'est pas toujours le cas que A soit *TU* ou que le système $Ax \leq b$ soit *TDE*. Ces remarques ne sont pas nécessairement négatives car ce dont nous avons besoin est d'avoir une bonne description autour de la solution optimale et une méthode efficace qui nous permet d'identifier si une inégalité donnée est violée ou non. i.e. Étant donnée une solution fractionnaire x^* , trouver un hyperplan séparant cette solution de l'enveloppe convexe ou montrer qu'un tel hyperplan n'existe pas en un temps polynomial, c'est ce qui est connu sous le nom de *problème de séparation* qui peut être défini formellement comme suit.

Définition 2.2. Étant donné un système d'inégalités linéaires $Ax \leq b$ et un vecteur x^* , le problème de séparation associé à $Ax \leq b$ est de trouver une inégalité de $Ax \leq b$ qui est violée par x^* ou décider que $Ax^* \leq b$.

Grötschel et al en 1981 ont établi une relation entre le problème d'optimisation et le problème de séparation qui lui est associé en utilisant l'algorithme de l'ellipsoïde. Cette relation implique que le problème d'optimisation et le problème de séparation qui lui est associé ont le même degré de difficulté.

Théorème 2.9 (Grötschel et al 1981). [45] Étant donné un programme linéaire $P = \max\{cx, Ax \leq b\}$, il existe un algorithme polynomial en n (où n est le nombre de variables) pour P si et seulement s'il existe un algorithme polynomial en n pour résoudre le problème de séparation associé à $Ax \leq b$.

D'après le théorème 2.9, la complexité d'un algorithme de coupes sur un polyèdre P dépend de la complexité du problème de séparation qui lui est associé et non du nombre de contraintes même s'il est exponentiel.

2.4 La technique de Lifting

La technique de lifting a été introduite pour la première fois par Gomory (1969) [41] et depuis elle présente un sujet de recherches privilégié. Elle a été suggérée notamment par Padberg [86], Wolsey [109], Zemel [112] et Balas et Zemel [6] et a été largement utilisée dans l'implémentation des algorithmes de plans coupants particulièrement par Crowder et al [23], Van Roy et Wolsey [91]. Cette technique est très utile pour l'extension des inégalités valides, elle consiste à étendre une inégalité valide en introduisant plus de variables dans l'inégalité. Selon la manière dont ces variables sont introduites, nous pouvons distinguer deux types de lifting :

- ◇ **Le lifting séquentiel** : Les variables sont introduites une par une séquentiellement et leurs coefficients dépendent de l'ordre dans lequel elles sont introduites. Il y a aussi le lifting séquentiel indépendant qui utilise la notion de fonctions superadditives [51] [110].
- ◇ **Le lifting simultané** : Un ensemble de variables est introduit à la fois.

2.4.1 Le lifting séquentiel

Généralement, c'est le type de lifting le plus utilisé. Les coefficients sont déterminés un par un. Nous commençons par une inégalité :

$$\sum_{j \in N_1} \pi_j x_j \leq \pi_0 \quad (2.12)$$

qui est valide pour S^0 (resp : définit une facette de $\text{conv}(S^0)$), où $N_1 \subseteq N = \{1, \dots, n\}$ est un sous ensemble d'indices de variables de N et $S^0 = \{x \in S \mid \forall j \in N \setminus N_1, x_j = 0\}$, où S est un ensemble discret de solutions. Nous résolvons le problème suivant :

$$Z^* = \max \left\{ \sum_{j \in N_1} \pi_j x_j : Ax \leq b, x_j = 1, x \in \{0, 1\}^n \right\}. \quad (2.13)$$

Si le problème (2.13) n'a pas de solution, alors $x_j = 0$ est valide pour S^1 , où $S^1 = \{x \in S \mid \forall j \in N \setminus N_1, x_j = 1\}$, sinon l'inégalité $\alpha_1 x_1 + \sum_{j \in N_1} \pi_j x_j \leq \pi_0$ est valide pour S^1 et ce pour tout $\alpha_1 \leq \pi_0 - Z^*$. En itérant ce processus pour toutes les variables $j \in N \setminus N_1$ qui seront introduites séquentiellement dans un certain ordre, nous obtenons l'inégalité :

$$\sum_{j \in N \setminus N_1} \alpha_j x_j + \sum_{j \in N_1} \pi_j x_j \leq \pi_0 \quad (2.14)$$

qui est valide pour S^1 (resp : définit une facette de $\text{conv}(S^1)$).

Exemple 2.1. Supposons que l'inégalité :

$$\sum_{j=2}^6 x_j \leq 2$$

est une facette de $\text{conv}(S^0)$, où $S^0 = \{x \in S \mid x_1 = 0\}$. Nous cherchons à déterminer la valeur du coefficient α , avec $\alpha > 0$, telle que :

$$\alpha x_1 + \sum_{j=2}^6 x_j \leq 2 \quad (2.15)$$

soit valide pour S .

Nous avons l'inégalité (2.15) qui est valide pour S^0 . Elle sera valide pour $S^1 = \{x \in S \mid x_1 = 1\}$ si et seulement si :

$$\alpha + \max_{x \in S^1} \sum_{j=2}^6 x_j \leq 2$$

Nous avons $\forall x \in S^1$, où $S^1 = \{x \in S \mid x_1 = 1\}$: $\sum_{j=2}^6 x_j = 0 \Rightarrow \alpha = 2$.

D'où : $2x_1 + \sum_{j=2}^6 x_j \leq 2$ est valide pour S .

2.4.2 Cas général du lifting séquentiel

Dans cette section nous allons présenter le principe de la technique de lifting. La proposition (2.10) suivante montre comment étendre une inégalité valide de S^0 à S .

Proposition 2.10. [84] *Supposons que $S \subseteq \{0, 1\}^n$, $S^\delta = S \cap \{x \in \{0, 1\}^n \mid x_1 = \delta\}$ pour $\delta \in \{0, 1\}$ et*

$$\sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (2.16)$$

est valide pour S^0 .

Si $S^1 = \emptyset$, alors $x_1 \leq 0$ est valide pour S .

Si $S^1 \neq \emptyset$, alors :

$$\alpha_1 x_1 + \sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (2.17)$$

est valide pour S pour tout $\alpha_1 \leq \pi_0 - \zeta$, où $\zeta = \max\{\sum_{j=2}^n \pi_j x_j : x \in S^1\}$. De plus, si $\alpha_1 = \pi_0 - \zeta$ et l'inégalité (2.16) induit une face de dimension k de $\text{conv}(S^0)$, alors l'inégalité (2.17) induit une face de dimension au moins $k + 1$ de $\text{conv}(S)$. (Si l'inégalité (2.16) induit une facette de $\text{conv}(S^0)$, alors l'inégalité (2.17) induit une facette de $\text{conv}(S)$)

Le principe du lifting est aussi applicable pour étendre une inégalité valide de S^1 à S , comme le montre la proposition suivante.

Proposition 2.11. [84] *Supposons que l'inégalité (2.16) est valide pour S^1*

Si $S^0 = \emptyset$, alors $x_1 \geq 1$ est valide pour S .

Si $S^0 \neq \emptyset$, alors :

$$\gamma_1 x_1 + \sum_{j=2}^n \pi_j x_j \leq \pi_0 + \gamma_1 \quad (2.18)$$

est valide pour S pour tout $\gamma_1 \geq \zeta - \pi_0$, où $\zeta = \max\{\sum_{i=2}^n \pi_i x_i : x \in S^0\}$. De plus, si $\gamma_1 = \zeta - \pi_0$ et (2.16) induit une face de dimension k de $\text{conv}(S^1)$, alors (2.18) induit une face de dimension au moins $k + 1$ de $\text{conv}(S)$.

Quand $\alpha_1 = \pi_0 - \zeta$ dans la proposition (2.10) ou quand $\gamma_1 = \zeta - \pi_0$ dans la proposition (2.11), nous dirons que *le lifting est maximal*. Les propositions (2.10) et (2.11) peuvent être utilisées séquentiellement.

2.4.3 Propriété du lifting séquentiel

Les coefficients $\{\alpha_j\}$ dans l'inégalité (2.14) sont dépendants de l'ordre dans lequel les variables sont liftées. Donc, par la considération des différents ordres de $N \setminus N_1$, nous pouvons avoir une famille d'inégalités valides pour S .

La proposition suivante montre comment le coefficient d'une variable dépend de l'ordre dans lequel elle est introduite.

Proposition 2.12. [84] *Soit $N \setminus N_1 = \{1, 2, \dots, t\}$. Supposons que quand la proposition (2.10) est appliquée séquentiellement en utilisant le lifting maximum dans l'ordre $(i_1, i_2, \dots, i_{k-1}, i_k, \dots, i_t)$, l'inégalité :*

$$\sum_{j \in N \setminus N_1} \alpha_j x_j + \sum_{j \in N_1} \pi_j x_j \leq \pi_0$$

est obtenue. Alors, pour tout ordre $(i'_1, i'_2, \dots, i'_{k-1}, i'_k, \dots, i'_t)$ avec $i'_j = i_j$ pour $j = 1, \dots, k-1$, l'inégalité résultante :

$$\sum_{j \in N \setminus N_1} \alpha'_j x_j + \sum_{j \in N_1} \pi_j x_j \leq \pi_0$$

obtenue par le lifting maximum a : $\alpha'_{i_k} \leq \alpha_{i_k}$.

2.4.4 Le lifting simultané

Contrairement au lifting séquentiel, le lifting simultané consiste à introduire simultanément un ensemble de variables. Étant donnée une inégalité :

$$\sum_{j \in N_1} \pi_j x_j \leq \pi_0 \tag{2.19}$$

qui est valide pour S^0 (resp : définit une facette de $\text{conv}(S^0)$), où $N_1 \subseteq N = \{1, \dots, n\}$ un sous ensemble d'indices de variables de N , $S^0 = \{x \in S \mid \forall j \in N_2, x_j = 0\}$ et $N_2 \subseteq N \setminus N_1$. Nous liftons simultanément le sous ensemble de variables N_2 , l'inégalité résultante :

$$\sum_{j \in N_2} \alpha_j x_j + \sum_{j \in N_1} \pi_j x_j \leq \pi_0 \tag{2.20}$$

est valide pour S^1 (resp : définit une facette de $\text{conv}(S^1)$).

Exemple 2.2. Soit l'inégalité :

$$\sum_{j=1}^4 x_j \leq 3$$

qui est une facette de $\text{conv}(S^0)$, où $S^0 = \{x \in S \mid \forall i \in N_2 : x_i = 0\}$ avec $N_2 = \{5, 6, 7, 8, 9, 10, 11\}$. En liftant simultanément toutes les variables de l'ensemble N_2 , nous obtenons l'inégalité :

$$\sum_{j=1}^4 x_j + \frac{1}{3} \sum_{j=5}^{11} x_j \leq 2$$

qui est valide pour S .

2.4.4.1 État de l'art sur le lifting simultané

En 1978, Zemel [112] a introduit la technique du lifting simultané pour un ensemble de variables entières. Sa méthode était restreinte aux variables binaires et permet de trouver toutes les facettes simultanément liftées en se basant sur la détermination des points extrêmes qui sont obtenus par la résolution de nombreux exponentiels programmes linéaires en nombres entiers. Depuis et pendant deux décennies, ce sujet de recherche n'a pas été exploré. Plus tard, vers 1999, beaucoup de recherches [51] [110] ont été faites par ce qui est connu comme le lifting séquentiel indépendant qui essaye de trouver un moyen pour élaguer l'inconvénient principal dans le lifting séquentiel, i.e. la dépendance des coefficients de l'ordre dans lequel les variables sont introduites, en utilisant la notion de fonctions superadditives [51] [110]. En 2005, Easton et Kevin [32] proposent un algorithme qui permet de lifter simultanément un ensemble de variables en une inégalité de couverture pour le problème du sac à dos binaire en un temps polynômial. Récemment, Sharma [98] a travaillé sur les perspectives de ces derniers auteurs, pour la détermination de la couverture minimale (C) et l'ensemble de variables à lifter(E), et a développé un algorithme nommé MSLA (Maximal Simultaneously Lifted Algorithm) qui permet de trouver des inégalités de couverture maximales simultanément liftées pour le problème du sac à dos binaire. Il y a aussi Gutierrez [53] qui a développé une technique pour lifter simultanément des ensembles de variables de nombres entiers.

2.4.5 Interprétation géométrique du lifting

Nous allons examiner le processus du lifting dans l'espace polaire :

$$\Pi^1 = \{\pi \in \mathbb{R}^n, \pi x \leq 1 \text{ pour tout } x \in S \subseteq \{0, 1\}^n\} \quad (2.21)$$

Si $\sum_{j=2}^n \pi_j x_j \leq 1$ est valide pour S^0 , le lifting maximal peut être décrit par un problème d'optimisation à une dimension dans l'espace polaire Π^1 :

$$\max\{\alpha : (0, \pi_2, \dots, \pi_n) + \alpha(1, 0, \dots, 0) \in \Pi^1\} \quad (2.22)$$

Pour interpréter le lifting séquentiel géométriquement, nous supposons que nous sommes dans la dimension 2 et que Π^1 a trois points extrêmes $\{\pi^0, \pi^1, \pi^2\}$.

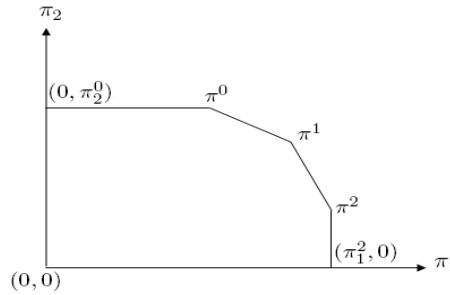


FIG. 2.1: Interprétation géométrique du lifting

Comme $\pi_2^0 > \max\{\pi_2^1, \pi_2^2\}$, nous avons $\pi_2^0 x_2 \leq 1$ qui donne une facette de $\text{conv}(S^0)$, où $S^0 = S \cap \{x \in \{0, 1\}^2 : x_1 = 0\}$. Le lifting maximum dans l'ordre (2, 1) donne une facette de $\text{conv}(S)$ définie par $\pi_1^0 x_1 + \pi_2^0 x_2 \leq 1$, ce qui est équivalent à se déplacer de $(0, \pi_2^0)$ dans la direction $(1 \ 0)$ pour obtenir le point extrême π^0 de Π^1 . D'une manière similaire, par un lifting maximum de $\pi_1^2 x_1 \leq 1$ dans l'ordre (1, 2), nous obtenons une facette de $\text{conv}(S)$ définie par $\pi_1^2 x_1 + \pi_2^2 x_2 \leq 1$, ce qui est équivalent à se déplacer de $(\pi_1^2, 0)$ dans la direction $(0 \ 1)$ pour obtenir le point extrême π^2 de Π^1 .

Nous voyons aussi qu'il n'y a pas une façon (chemin) de générer une facette de $\text{conv}(S)$ définie par $\pi_1^1 x_1 + \pi_2^1 x_2 \leq 1$ par un lifting séquentiel. En principe, le lifting n'est pas restreint à choisir un coefficient. Si nous observons que le lifting séquentiel maximal est équivalent à trouver un point extrême dans un polyèdre à une dimension, il n'est pas surprenant que dans le lifting simultané de k coefficients, le meilleur lifting est obtenu en trouvant les points extrêmes d'un polyèdre de dimensions k . Par conséquent, si nous commençons par l'inégalité $0 \leq 1$ et en liftons simultanément (π_1, π_2) , nous pouvons, en effet, obtenir les trois points extrêmes : π^0, π^1 et π^2 .

Le problème du sac à dos

Le problème du sac à dos est le problème NP-Difficile le plus simple en optimisation combinatoire, car il maximise une fonction objectif sujette à une simple contrainte de capacité. Le problème du sac à dos consiste à remplir un sac de capacité d en utilisant une collection de n objets. Chaque objet j possède un profit p_j et un poids w_j . L'objectif est de maximiser le profit total des objets à mettre dans le sac de sorte que leurs poids total n'excède pas la capacité d du sac. Mais quels sont les objets que nous devons mettre dans le sac ? C'est un problème qui est simple à formuler mais difficile à résoudre !

Dans ce chapitre, nous allons aborder le problème du sac à dos et plus précisément le cas classique (le problème du sac à dos en 0 – 1). Nous commençons d'abord par une présentation générale du problème qui inclue la formulation du problème, l'historique du problème, les propriétés et quelques applications existantes ainsi que la complexité du problème. Par la suite, nous allons énumérer plusieurs variantes de ce problème. Puis, nous allons aborder les méthodes de résolution exactes, approximatives et heuristiques. Et nous terminons par une exploration du polytope du problème de sac à dos.

3.1 Présentation du problème du sac à dos

3.1.1 Formulation du problème

Le problème du sac à dos binaire est l'un des problèmes d'optimisation combinatoire le plus largement étudié, car il apparaît comme sous problème (sous structure) de plusieurs problèmes qui comportent une contrainte de capacité ou de budget.

Étant donné un ensemble de n objets et un sac de capacité d , où chaque objet j a un poids w_j et un profit p_j , le problème consiste à sélectionner un sous ensemble d'objets qui maximise le profit total correspondant tout en n'excédant pas la capacité d du sac. Ce problème peut être formulé mathématiquement sous la forme d'un problème de maximisation :

$$\left\{ \begin{array}{l} \max \quad Z = \sum_{j=1}^n p_j x_j \\ \sum_{j=1}^n w_j x_j \leq d \\ x_j = \{0, 1\} \quad j = 1..n \end{array} \right. \quad (3.1)$$

où

$$x_j = \begin{cases} 1 & \text{si l'objet } j \text{ est sélectionné;} \\ 0 & \text{sinon.} \end{cases} \quad (3.2)$$

Nous supposons que toutes les valeurs p_j et w_j sont des nombres entiers positifs, $w_j < d$, $j = 1..n$ et $\sum_{j=1}^n w_j > d$.

3.1.2 Historique

Dans *les années cinquante*, la théorie de la *programmation dynamique* de Bellman [11] a produit le premier algorithme pour la résolution exacte du problème du sac à dos binaire. En 1957, Dantzig [28] a proposé une élégante et efficace méthode pour la détermination de la solution du problème relaxé et il a dérivé *une borne supérieure* pour la valeur de la solution entière. Cette borne est utilisée dans la vingtaine d'années suivantes dans la plupart des études consacrées au problème du sac à dos.

Dans *les années soixante*, la méthode *Branch and Bound* est développée. Elle s'est montrée capable de résoudre des problèmes avec un très grand nombre de variables. L'algorithme le plus connu à cette époque était celui de Horowitz et Sahni [58].

En 1973, *Ingariola et Korsh* [60] présentent la première *procédure de réduction*. En 1974, *Johnson* [61] a donné le premier *schéma d'approximation polynômial (PTAS)* pour le problème de somme de sous-ensembles. Ces résultats sont étendus par *Sahni* [92] pour le problème du sac à dos en 0 – 1. En 1975, le premier *schéma d'approximation totalement polynômial (FPTAS)* est obtenu par *Ibarra et Kim* [59]. En 1977, *Martello et Toth* [76] proposent la première *borne supérieure* qui domine la borne de Dantzig.

Les résultats principaux *des années quatre-vingt* concernent la résolution des *problèmes de grande taille*, pour lesquels le tri des variables consomme le plus grand pourcentage du temps d'exécution. En 1980, *Balas et Zemel* [7] présentent une nouvelle approche pour la résolution du problème du sac à dos par un tri qui se restreint à un petit sous ensemble de variables (*le problème du noyau*). Plusieurs algorithmes ont été conçus en se basant sur cette idée.

Les recherches des années *quatre vingt dix* sont concentrées sur la résolution des *instances difficiles de taille raisonnable* au lieu des instances extrêmement larges et faciles. Ces recherches ont apporté des algorithmes avec une amélioration de la borne du temps d'exécution dans le pire des cas. En 1995, *Pisinger* [88] a été le premier à présenter une récursive de la programmation dynamique pour le problème de somme de sous-ensembles qui a une complexité dans le pire des cas nettement meilleure que la récursive classique de Bellman. En 1997, *Martello et Toth* [79] ont montré comment les bornes supérieures dérivées à partir de la relaxation lagrangienne peuvent aider à résoudre plusieurs problèmes difficiles.

Chaque année, de nombreux articles sur le problème du sac à dos paraissent, plusieurs nouvelles variantes du problème classique du sac à dos sont considérées, nous citons à titre d'exemple : le problème du sac à dos multiple, quadratique, multidimensionnel ... Les techniques utilisées pour ces problèmes sophistiqués varient, mais il est intéressant de mentionner que beaucoup d'idées du problème du sac à dos binaire sont appliquées même dans les versions généralisées.

3.1.3 Quelques applications du problème du sac à dos

En parcourant l'histoire du problème du sac à dos, nous pouvons se demander quelle importance a ce problème pour avoir reçu toute cette attention? Tout simplement, parce que c'est un problème qui a de nombreuses applications aussi bien dans le domaine pratique que théorique.

Le problème du sac à dos apparaît dans de nombreuses applications en industrie et dans le domaine des finances, nous citons à titre d'exemple :

- ◇ Gestion financière (Cord (1964) [22], Kaplan (1966) [63], Hansmann(1961) [55]) ;
- ◇ Chargement de cargaisons (Bellman et Drefuls (1962) [12]) ;
- ◇ Problèmes de découpe (Gilomre et Gomory (1963, 1966) [38]) ;
- ◇ Tolérance aux fautes (Sinha et Zottners (1979) [99]) ;
- ◇ Cryptosystèmes (Diffé et Hellman (1976) [30]).

Le problème du sac à dos a aussi un intérêt théorique qui se résume dans les deux points suivants :

1. Dans de nombreuses applications, un problème de programmation linéaire en nombres entiers peut être transformé en un problème de sac à dos (Salkin (1975) [94], Syslo et al (1983) [102]) ;
2. Dans la résolution de nombreux programmes linéaires en nombres entiers, le problème du sac à dos apparaît comme un sous problème.

3.1.4 Propriétés du problème du sac à dos

- ★ **La borne supérieure :** Une propriété importante du problème du sac à dos est qu'une borne supérieure peut être obtenue à partir de la solution optimale du problème relaxé continu. En 1957, Dantzig [28] a établi une manière simple pour déterminer une solution optimale du problème du sac à dos continu $C(KP)$ et a dérivé la première *borne supérieure*.

$$C(KP) \left\{ \begin{array}{l} \max \quad Z = \sum_{j=1}^n p_j x_j \\ \sum_{j=1}^n w_j x_j \leq d \\ 0 \leq x_j \leq 1 \quad j = 1..n \end{array} \right. \quad (3.3)$$

Dantzig a utilisé un algorithme glouton pour remplir le sac à dos, les étapes suivies peuvent être résumées de la manière suivante :

- Ordonner les objets selon le ratio profit/poids :

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \dots \geq \frac{p_n}{w_n} \quad (3.4)$$

- A chaque étape, nous choisissons un objet qui a le plus grand ratio profit/poids, jusqu'à atteindre le premier objet qui ne peut pas être mis dans le sac à dos. Cet

objet et dénoté par objet critique b :

$$b = \min \left\{ j \mid \sum_{i=1}^j w_i > d \right\} \quad (3.5)$$

- La solution optimale de $C(KP)$ est :

$$x_i^* = \begin{cases} 1, & i = 1..b-1 \\ (d - \sum_{i=1}^{b-1} w_i) / w_b, & i = b \\ 0, & i = b+1..n \end{cases} \quad (3.6)$$

- La valeur de la solution optimale de $C(KP)$ est :

$$Z = \sum_{i=1}^{b-1} p_i + (d - \sum_{i=1}^{b-1} w_i) \frac{p_b}{w_b} \quad (3.7)$$

Cette solution optimale donne immédiatement lieu à la borne supérieure suivante :

$$U_1 = \sum_{i=1}^{b-1} p_i + \lfloor (d - \sum_{i=1}^{b-1} w_i) \frac{p_b}{w_b} \rfloor \quad (3.8)$$

Cette borne est appelée *borne de Dantzig*.

L'existence d'une borne supérieure très forte et que nous pouvons obtenir rapidement rend possible le développement d'algorithmes par énumération implicite efficaces pour la résolution des problèmes à l'optimum, ce qui a incité les chercheurs à améliorer cette borne en utilisant différentes techniques [78] [65] :

- ✓ Borne par des contraintes additionnelles (Martello et Toth(1997) [79], Fayard et Plateau(1982) [35]) ;
- ✓ Borne par la relaxation lagrangienne (Müller et Merbach(1978) [82], Dudzinski et Walukiewicz(1984) [31]) ;
- ✓ Borne par énumération partielle (Martello et al(1988) [77]) ;
- ✓ Borne par des inégalités valides (Martello et al(1997) [79]).

- ★ **Le problème du noyau** : Pour une résolution efficace du problème du sac à dos, la plupart des algorithmes exigent le tri des objets selon l'ordre décroissant du ratio profit/poids. Balas et Zemel (1980) observent que la majorité du temps de calcul pour la résolution des grands problèmes est prise par le procédé du tri. Néanmoins, durant le processus de résolution du problème du sac à dos, il a été remarqué que, ayant résolu le problème relaxé continu, en général, seulement quelques variables de décision ont besoin d'être changées pour obtenir une solution optimale entière. Ces variables se trouvent autour de l'élément critique et

le sous ensemble d'objets qu'elles représentent est appelé *noyau*. Donc, pour éviter un tri complet, nous pouvons trier seulement les objets dans le noyau.

- ★ **Le problème du sac à dos est NP-difficile dans le sens faible :** Actuellement, nous savons que les problèmes (3.1), (3.10), (3.19), (3.21), (3.22) et (3.23) sont pseudo-polynomiaux, car ce sont des problèmes NP-difficiles dans le sens faible, d'où, nous pouvons peut leurs trouver des schémas d'approximation totalement polynomiaux (*FPTAS*). Tandis que le reste des variantes sont NP-difficiles dans le sens fort, cela signifie que des algorithmes pseudo-polynomiaux ne peuvent être conçus car sinon $NP = P$.
- ★ **Le problème du sac à dos est séparable :** Une autre propriété importante du problème du sac à dos est qu'il est séparable ; i.e. Nous pouvons résoudre un problème du sac à dos en divisant les objets en deux ensembles. Puis, énumérer toutes les solutions faisables dans chaque ensemble. Et enfin, fusionner les deux ensembles de solutions faisables. Cette propriété a été observée par Horowitz et Sahni (1974) [58]. Une conséquence de cette observation est que nous pouvons résoudre un problème du sac à dos en $0-1$ par un calcul parallèle en divisant le problème récursivement en deux parties. L'inconvénient est que le nombre de processeurs requis est énorme.

3.1.5 Complexité du problème du sac à dos

Le problème du sac à dos est un problème NP-difficile, sa version décision présentée ci-après est NP-complet.

Version décision du problème de sac à dos :

Problème : Sac à Dos

Données : $(u_1 \ u_2 \ \dots \ u_n)$, $(w_1 \ w_2 \ \dots \ w_n) \in \mathbb{Z}^n$, $\beta, k \in \mathbb{Z}$.

Question : Existe-il une affectation des variables x_1, x_2, \dots, x_n à $0-1$ tel que :

$$\left\{ \begin{array}{l} w_1 x_1 + w_2 x_2 + \dots + w_n x_n \geq k; \\ \text{et} \\ u_1 x_1 + u_2 x_2 + \dots + u_n x_n \geq k. \end{array} \right. \quad (3.9)$$

Il y a un lien entre *la version décision* et *la version optimisation* du problème dans la mesure où s'il existe un algorithme polynomial qui résout la version décision, alors nous pouvons trouver la valeur optimale pour le problème d'optimisation en temps polynomial, en appliquant itérativement cet algorithme tout en augmentant la valeur de k . D'une manière similaire, si un algorithme

trouve la valeur optimale du problème d'optimisation en temps polynomial, alors le problème de décision peut être résolu en temps polynomial en comparant la valeur de la solution donnée par cet algorithme avec k . Ainsi, les deux versions du problème sont de difficulté similaire.

3.2 Variantes et extensions du problème du sac à dos

3.2.1 Le problème sac à dos en 0 – 1

3.2.1.1 Le problème sac à dos à choix multiple

Le problème du sac à dos à choix multiple est un problème du sac à dos en 0-1 dans lequel une partition N_1, N_2, \dots, N_r des objets est donnée. La formulation du problème est :

$$\left\{ \begin{array}{l} \max \quad Z = \sum_{j=1}^n p_j x_j \\ \sum_{j=1}^n w_j x_j \leq d \quad i = 1..m \\ \sum_{j \in N_k} x_j = 1 \quad k = 1..r \\ x_j = 0 \text{ ou } 1, \quad j \in N = \{1, \dots, n\} = \bigcup_{k=1}^r N_r \end{array} \right. \quad (3.10)$$

Nous supposons que :

$$N_h \cap N_k = \emptyset \text{ pour tout } h \neq k$$

Le problème est *NP*-difficile au sens faible, puisque chaque instance du problème du sac à dos, ayant r éléments de profits p_j et de poids w_j ($j = 1, \dots, r$) et un sac de capacité d , est équivalente à une instance du problème du sac à dos à choix multiple obtenue en posant $n = 2r$, $p_j = w_j = 0$, pour $j = r + 1, \dots, 2r$ et $N_k = \{k, r + k\}$, pour $k = 1..r$.

3.2.1.2 Le problème du sac à dos multidimensionnel

Nous considérons ici que le sac à dos a y dimensions, avec $y > 0$. Par exemple, nous pouvons imaginer une boîte. Chaque objet a trois dimensions, et il ne faut pas déborder sur aucune des

dimensions. Formellement, le problème du sac à dos multidimensionnel est le suivant :

$$\left\{ \begin{array}{l} \max \quad Z = \sum_{j=1}^n p_j x_j \\ \\ \sum_{j=1}^n w_{ij} x_j \leq d_i \quad i = 1..y \\ \\ x_j \in \{0, 1\} \quad j = 1..n \end{array} \right. \quad (3.11)$$

Nous mentionnons que le problème du sac à dos multidimensionnel est *NP*-difficile au sens fort. En pratique, la version multidimensionnelle peut servir à modéliser et à résoudre le problème du remplissage d'un container dont le volume et la charge maximale sont limités.

3.2.1.3 Le problème du sac à dos multiple

Considérons le problème où nous avons à répartir un ensemble de n objets dans m sacs à dos ($m \leq n$) de capacités différentes, avec :

- p_j : Le profit de l'objet j ;
- w_j : Le poids de l'objet j ;
- d_i : La capacité du sac i .

Le problème est de sélectionner m sous ensembles disjoints d'objets, tel que, le sous ensemble i s'emboîte dans le sac de capacité d_i et la valeur totale des objets sélectionnés est maximale. Formellement, nous pouvons définir le problème du sac à dos multiple comme le programme linéaire en nombres entiers suivant :

$$\left\{ \begin{array}{l} \max \quad Z = \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} \\ \\ \sum_{j=1}^n w_j x_{ij} \leq d_i, \quad i = 1..m \\ \\ \sum_{i=1}^m x_{ij} \leq 1, \quad j = 1..n \\ \\ x_{ij} \in \{0, 1\}, \quad i = 1..m, \quad j = 1..n \end{array} \right. \quad (3.12)$$

où

$$x_{ij} = \begin{cases} 1, & \text{Si l'objet } i \text{ est assigné au sac } j; \\ 0, & \text{Sinon.} \end{cases} \quad (3.13)$$

Nous allons supposer que les coefficients p_j , w_j et d_i sont des entiers positifs. Nous allons aussi supposer que :

$$\max_{j=1..n} w_j \leq \max_{i=1..m} d_i \quad (3.14)$$

$$\min_{i=1..m} d_i \geq \min_{j=1..n} w_j \quad (3.15)$$

$$\sum_{j=1}^n w_j \geq \max_{i=1..m} d_i \quad (3.16)$$

La première hypothèse garantit que chaque objet s'emboîte dans au moins un sac à dos sinon elle peut être retirée du problème. Si la seconde inégalité est violée, nous pouvons ne pas tenir compte du plus petit sac à dos, car aucun objet n'entre dedans. Finalement, la dernière inégalité permet d'échapper à la solution triviale où tous les objets s'emboîtent dans le plus grand sac à dos.

Le problème du sac à dos multiple est connu d'être un problème appartenant à la classe NP -difficile, puisque dans le cas où le nombre de sacs est égal à un, cela coïncide avec le problème du sac à dos en 0 – 1 qui est NP -difficile.

3.2.1.4 Le problème du sac à dos quadratique

Étant donné un ensemble de n objets $N = \{1, \dots, n\}$ de poids entiers positifs et un sac de capacité d . Le profit des objets est donné par la matrice entière non négative $P = (p_{ij})$, où $q_j = p_{jj}$ est le profit obtenu si l'objet j est sélectionné (Il dénote la diagonale de la matrice P), et pour $j > i$ $p_{ij} + p_{ji}$ est le profit obtenu si les deux objets i et j sont sélectionnés. Le problème du sac à dos quadratique consiste à sélectionner un sous ensemble d'objets dont la somme des poids ne dépasse pas la capacité du sac, de tel sorte que la somme des valeurs soit maximale. Le problème a la formulation mathématique suivante :

$$\left\{ \begin{array}{l} \max \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j \\ \sum_{j \in N} w_j x_j \leq d \\ x_j \in \{0, 1\}, \quad j \in N \end{array} \right. \quad (3.17)$$

où

$$x_j = \begin{cases} 1 & \text{si l'objet } j \text{ est sélectionné;} \\ 0 & \text{sinon.} \end{cases} \quad (3.18)$$

Nous supposons que $\max_{j \in N} w_j \leq d < \sum_{j \in N} w_j$ et que la matrice des valeurs P est symétrique, i.e., $p_{ij} = p_{ji}$, pour tout $i, j \in N$ et $j > i$.

Le problème du sac à dos quadratique est une généralisation du problème du sac à dos qui apparaît quand $p_{ij} = 0$, pour tout $i \neq j$.

3.2.1.5 Le problème de somme de sous-ensembles

Le problème de somme de sous-ensembles (en anglais : subset sum) appelé aussi le problème du sac à dos à valeurs indépendantes, est un cas particulier du problème du sac à dos en 0-1 qui apparaît quand les profits et les poids des objets sont identiques ($p_j = w_j$).

Ayant un ensemble de n objets N de poids w_j et un sac de capacité d , le problème consiste à trouver un sous ensemble d'objets dont la somme totale des poids est la plus proche de la capacité d du sac sans l'excéder. La formulation mathématique du problème est :

$$\left\{ \begin{array}{l} \max \sum_{j=1}^n w_j x_j \\ \sum_{j=1}^n w_j x_j \leq d \\ x_j \in \{0, 1\} \quad j \in N = \{1, \dots, n\} \end{array} \right. \quad (3.19)$$

où

$$x_j = \begin{cases} 1 & \text{si l'objet } j \text{ est sélectionné;} \\ 0 & \text{sinon.} \end{cases} \quad (3.20)$$

Le problème de somme de sous-ensembles est un problème important dans le domaine de la cryptographie. Il est utilisé dans plusieurs systèmes de génération de clés publiques.

3.2.2 Le problème du sac à dos en variables entières

Dans le problème de sac à dos en variables entières, nous considérons que nous avons plusieurs exemplaires de chaque type d'objets. Le problème consiste donc à trouver le nombre d'exemplaires à prendre pour chaque type d'objets :

3.2.2.1 Le problème du sac à dos borné

Si nous avons une borne m_j de la quantité d'objets de type j , alors nous parlons de problème du sac à dos borné qui peut être formulé comme suit :

$$\left\{ \begin{array}{l} \max \quad Z = \sum_{j=1}^n p_j x_j \\ \sum_{j=1}^n w_j x_j \leq d \\ x_j = \{0, 1, \dots, m_j\} \quad j = 1..n \end{array} \right. \quad (3.21)$$

x_j : représente la quantité d'objets de type j à inclure dans le sac de sorte à maximiser la valeur de la fonction objectif.

3.2.2.2 Le problème du sac à dos non borné

C'est une généralisation du problème du sac à dos borné, où nous disposons d'un nombre illimité d'objets pour chaque type. La formulation du problème est :

$$\left\{ \begin{array}{l} \max \quad Z = \sum_{j=1}^n p_j x_j \\ \sum_{j=1}^n w_j x_j \leq d \\ x_j \geq 0 \text{ entier} \quad j = 1..n \end{array} \right. \quad (3.22)$$

x_j : va être borné par la capacité d du sac à dos, comme le poids de chaque objet est au moins égal à un.

3.2.2.3 Le problème de Change-Making

Formellement, le problème Change-Making peut être défini comme suit. Étant donné un ensemble de n types d'objets et un sac de capacité d , avec w_j est le poids de l'objet de type j , le problème consiste à sélectionner un nombre x_j ($j = 1..n$) d'objets de chaque type tel que le poids total soit égal à d et le nombre d'objets soit minimum. La formulation mathématique du problème est la suivante :

$$\left\{ \begin{array}{l} \min \quad Z = \sum_{j=1}^n x_j \\ \sum_{j=1}^n w_j x_j = d \\ x_j \geq 0 \text{ et entier} \quad j \in N = \{1, \dots, n\} \end{array} \right. \quad (3.23)$$

Nous supposons que les w_j et d sont des entiers et que :

$$w_j < d \quad j \in N \quad (3.24)$$

$$w_i \neq w_j \quad \text{si } i \neq j \quad (3.25)$$

Le problème Change-Making est *NP*-difficile au sens faible et peut s'interpréter comme une caissière chargée d'assembler une somme donnée d'argent en utilisant un nombre minimum de pièces de valeurs précisées dans la caisse, où le nombre de pièces disponible illimité. Il peut aussi être vu comme un problème du sac à dos non borné, où $p_j = -1$, pour tout j , et une égalité stricte est imposée à la contrainte de capacité.

3.2.3 Le problème du sac à dos en variables continues

Le problème du sac à dos en variables continues est obtenu en relâchant la contrainte d'intégrité sur les variables, c'est-à-dire, que nous autorisons à ne prendre qu'une fraction des objets dans le sac à dos. Le problème du sac à dos en variables continues appartient à la classe *P*.

3.3 Approches de résolution

3.3.1 Approches exactes

De nombreux algorithmes exacts ont été développés pour le problème du sac à dos, ceux-ci sont basés sur :

- ◇ L'approche par énumération implicite ;
- ◇ La programmation dynamique.

3.3.1.1 Approche par énumération implicite

Dans les dernières décennies, de nombreux algorithmes pour la résolution du problème du sac à dos, basés sur l'approche par énumération implicite, ont été proposés. Les différences entre ces algorithmes résident dans le principe de séparation, le principe d'élagage et la stratégie de développement de l'arborescence. La plupart des algorithmes efficaces de l'approche par énumération implicite sont basés sur la stratégie de développement de l'arborescence de branchement profondeur d'abord, nous citons :

- Kolesar (1967) [66] était le premier à présenter un algorithme pour la résolution du problème du sac à dos en 0 – 1 par l'approche par énumération implicite. L'algorithme recherche une meilleure solution en se branchant du noeud faisable pour lequel la borne de Dantzing est maximale ;
- Greenberg et Hegerich (1970) [42] ont développé l'algorithme de Kolesar en utilisant la stratégie profondeur d'abord ;
- Horowitz et Sahni (1974) [58] ont présenté des améliorations. Ils ont appliqué la stratégie profondeur d'abord avec le même choix de la variable de branchement que l'algorithme de Kolesar ;
- Martello et Toth (1977) [76] ont développé une approche par énumération implicite plus efficace ;
- Pisinger (1995) [88] a développé un algorithme qu'il a nommé *primal-dual algorithm*.

3.3.1.2 La programmation dynamique

La programmation dynamique consiste à plonger le problème proposé dans un problème plus général, puis le résoudre par récurrence sur des paramètres dont il est dépendant. Le problème initial correspond à une valeur donnée des paramètres.

A la différence de l'approche par énumération implicite, qui s'applique lorsqu'il est possible de diviser le problème en sous problèmes indépendants, la programmation dynamique est envisagée lorsque :

- La subdivision n'est pas facile à déterminer de façon optimale ;
- Les sous problèmes ne sont pas indépendants.

Le principe d'optimalité adopté par la programmation dynamique est que *toute sous-politique d'une politique optimale est optimale*, d'où, son cadre d'application est restreint aux problèmes qui vérifient ce principe.

Le terme programmation dynamique a été choisi par Bellman qui a introduit cette méthode, dans les années cinquante, dans un souci de communication. Son supérieur ne supportait ni le mot *recherche* ni celui de *mathématique*, alors il lui a semblé que les termes *programmation* et *dynamique* donnaient une apparence qui plairait à son supérieur. A l'époque, ce terme signifie beaucoup plus planification et ordonnancement et non la programmation au sens que nous lui donnons de nos jours.

Le premier algorithme exact basé sur la programmation dynamique pour la résolution du problème du sac à dos en 0 – 1 a été proposé par Bellman (1954, 1957) [11].

Pour un problème de grande taille, la technique de Bellman produit, dans chaque itération, une quantité énorme d'états qui doit être enregistrée. Des améliorations ont été apportées sur la technique de la programmation dynamique pour la résolution du problème du sac à dos en 0 – 1, nous citons :

- ✦ Horowitz et Sahni (1974) [58] ont présenté un modèle de programmation dynamique alternatif.
- ✦ Toth (1980) [103] a amélioré la récursive de Bellman.

3.3.2 Approches approximative

Comme le problème du sac à dos est *NP*-difficile, quelques instances peuvent être impossibles à résoudre à l'optimalité dans un temps raisonnable. Dans de telles situations, nous pouvons être intéressé par une solution approximative avec une valeur de la fonction objectif z , où l'erreur relative est bornée par une certaine constante ϵ , i.e.

$$\frac{z - z^*}{z^*} \leq \epsilon \quad (3.26)$$

où, z^* est la valeur optimale de la fonction objectif.

3.3.2.1 Schémas d'approximation totalement polynomiaux (FPTAS)

Le problème du sac à dos est un problème qui admet un schéma d'approximation totalement polynomial (*FPTAS*), car il est *NP*-difficile au sens faible.

En 1975, Ibarra et Kim [59] ont présenté le premier schéma d'approximation totalement polynomial (*FPTAS*) pour le problème du sac à dos en 0 – 1. Ainsi, pour tout $\epsilon > 0$, l'algorithme trouve une solution approximative z avec une erreur relative au plus égale à ϵ , tel que la complexité en

temps et en espace se développe polynomialement avec n et $\frac{1}{\epsilon}$, ($O(n/\epsilon^2)$). L'algorithme d'Ibarra et de Kim est basé sur la programmation dynamique.

Quelques années plus tard, plusieurs schémas d'approximations totalement polynomiaux (FP-TAS) ont été développés par différents auteurs. Un résumé de ces travaux est présenté dans le tableau suivant :

Auteur	temps d'exécution	espace mémoire
Ibarra et Kim [59]	$O(n \log n + (\frac{1}{\epsilon^2}) \cdot \min\{(\frac{1}{\epsilon^2}) \log(\frac{1}{\epsilon}), n\})$	$O(n + (\frac{1}{\epsilon^3}))$
Lawler [70]	$O(n \log(\frac{1}{\epsilon}) + (\frac{1}{\epsilon^4}))$	$O(n + (\frac{1}{\epsilon^3}))$
Magazine et Oguz [72]	$O(n^2 \log n - (\frac{1}{\epsilon}))$	$O(n - (\frac{1}{\epsilon}))$
Kellerer et Pferschy	$O(n \min\{\log n, \log(\frac{1}{\epsilon})\} + (\frac{1}{\epsilon^2}) \log(\frac{1}{\epsilon}) \cdot \min\{n, (\frac{1}{\epsilon} \log(\frac{1}{\epsilon}))\})$	$O(n + (\frac{1}{\epsilon^2}))$

TAB. 3.1: La complexité des schémas d'approximation totalement polynomiaux (FPTAS) pour le problème du sac à dos

3.3.3 Approches heuristiques

Parmi les heuristiques développées pour la résolution approchée du problème du sac à dos, nous trouvons les algorithmes gloutons. Dans ce qui suit, nous présentons une version simple de ces algorithmes qui consiste à :

- **Étape 1** : Ordonner les objets selon le ratio profit/poids qui est donné par la formule (3.4) ;
- **Étape 2** : A chaque étape, un élément, selon l'ordre précédemment défini, est sélectionné ;
- **Étape 3** : Tester si l'élément sélectionné est admissible ou non. Un élément est dit admissible, si son poids ne dépasse pas la capacité résiduelle (capacité restante après fixation des autres éléments) du sac, et non admissible dans le cas contraire ;
- **Étape 4** : Si l'élément sélectionné est admissible, alors il est mis dans le sac, sinon aller à l'étape 2 (sélectionner l'élément qui se situe juste après).

Ces étapes sont répétées jusqu'à l'épuisement des éléments admissibles. Ainsi, le sac à dos est rempli de proche en proche (d'une manière gloutonne).

Algorithme 1 : Algorithme glouton pour le problème de sac à dos

Données : Une instance d'un problème du sac à dos ;

Résultats : Une solution réalisable \bar{x} ;

```

 $\bar{d} := d$  ;
for  $j = 1$  to  $n$  do
  if  $w_j \leq \bar{d}$  then
     $\bar{x}_j := 1$  ;
     $d := \bar{d} - w_j$  ;
  else
     $\bar{x}_j := 0$  ;
  end
end
Sortir avec une solution réalisable  $\bar{x}$  ;

```

3.4 Le polytope du problème de sac à dos

Depuis le début des années 70, le polytope du sac à dos en 0 – 1 a été étudié par plusieurs de chercheurs, à savoir Balas [5], Hammer, Johnson et Peled [54], Wolsey [107] [108], et Balas et Zemel [6]. Récemment, la généralisation du polytope du problème de sac à dos en 0 – 1 a été intensivement étudiée par Weismantel [106].

Considérons l'ensemble des contraintes du problème du sac à dos en 0 – 1 :

$$\mathbb{P} = \left\{ x \in \{0, 1\}^n, \sum_{j \in N} w_j x_j \leq d, N = \{1, \dots, n\} \right\} \quad (3.27)$$

où, $w_j \in \mathbb{Z}_+^1$ pour $j \in N$, et $d \in \mathbb{Z}_+^1$.

Le polytope du problème de sac à dos \mathcal{P} est l'enveloppe convexe de l'ensemble de tous les vecteurs solutions pour le problème du sac à dos, c.-à-d.

$$\mathcal{P} = \text{conv} \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n w_j x_j \leq d, x_j \in \{0, 1\}, j = 1..n \right\} \quad (3.28)$$

Soit e^j le $j^{\text{ème}}$ vecteur unité, i.e. $e_i^j = 1$ et $e_i^j = 0$ pour $i \neq j$. En raison de l'hypothèse $w_j \leq d$ sur les poids des objets, les éléments de l'ensemble $V = \{e^1, \dots, e^n, 0\}$ sont des vecteurs solutions affinement indépendants du problème du sac à dos. Par conséquent, \mathcal{P} a une pleine dimension $\dim(\mathcal{P}) = n$. Les inégalités définissant des facettes simples de \mathcal{P} sont les inégalités $x_j \geq 0$ pour $j = 1..n$, ceci peut être vu par l'enlèvement de e^j de V , i.e. tous les vecteurs restants sont des éléments de $F = \{x \in \mathbb{P} \mid x_j = 0\}$ et sont affinement indépendants.

3.4.1 Inégalités de couverture

L'étude des facettes du polytope du problème de sac à dos remonte à Balas [5], Hammer, Johnson et Peled [54] et Wolsey [110] qui ont étudié une classe importante des inégalités valides pour le polytope du sac à dos. Ces inégalités sont connues sous le nom de inégalités de couverture.

Définition 3.1. Soit $C \subseteq N$ un sous-ensemble de N , C est appelé une couverture ou ensemble dépendant, pour \mathbb{P} si :

$$\sum_{j \in C}^n w_j > d \quad (3.29)$$

Définition 3.2. C est une couverture minimale pour \mathbb{P} si :

$$\sum_{j \in C \setminus \{i\}}^n w_j \leq d \quad (3.30)$$

pour tout $i \in C$.

Ceci signifie que la suppression de n'importe quel objet simple i de C va engendrer la perte de la propriété de couverture.

Proposition 3.1. [84] Si C est une couverture, alors :

$$\sum_{j \in C}^n x_j \leq |C| - 1 \quad (3.31)$$

est une inégalité valide pour \mathcal{P} .

Exemple 3.1. Soit le problème du sac à dos suivant [84] :

$$\mathbb{P} = \{x \in \{0, 1\}^5 : 79x_1 + 53x_2 + 53x_3 + 45x_4 + 45x_5 \leq 178\}$$

Les couvertures minimales et les inégalités valides correspondantes sont :

$$C_1 = \{1, 2, 3\} : x_1 + x_2 + x_3 \leq 2 \quad (3.32)$$

$$C_2 = \{1, 2, 4, 5\} : x_1 + x_2 + x_4 + x_5 \leq 3 \quad (3.33)$$

$$C_3 = \{1, 3, 4, 5\} : x_1 + x_3 + x_4 + x_5 \leq 3 \quad (3.34)$$

$$C_4 = \{2, 3, 4, 5\} : x_2 + x_3 + x_4 + x_5 \leq 3 \quad (3.35)$$

Proposition 3.2. Soit C une couverture minimale. L'inégalité minimale de couverture :

$$\sum_{j \in C}^n x_j \leq |C| - 1 \quad (3.36)$$

définit une facette pour le polytope du problème de sac à dos \mathcal{P}_C .

Preuve. On dénote par $\mathcal{P}_{\bar{N}}$ le polytope du problème du sac à dos défini pour l'ensemble des objets $\bar{N} \subseteq N$. Pour $s \in C$, soit $x^s \in \mathbb{R}^{|\bar{N}|}$ défini comme suit :

$$x_j^s = \begin{cases} 0, & j = s; \\ 1, & \text{sinon.} \end{cases} \quad (3.37)$$

Clairement, les vecteurs x^s , $s \in C$, sont affinement indépendants. Ainsi, l'inégalité minimale de couverture définit une facette du polytope du problème de sac à dos \mathcal{P}_C . \square

Généralement, les inégalités minimales de couverture ne sont pas des inégalités définissant des facettes pour \mathcal{P} , mais elles peuvent être liftées pour définir des facettes.

3.4.1.1 L'inégalité de couverture étendue

Un exemple pour une inégalité liftée de couverture est obtenu, en présentant l'extension $E(C)$ pour une couverture minimale C .

Définition 3.3. Soit C une couverture minimale, l'extension $E(C)$ de C est donnée par :

$$E(C) = C \cup \{ j \in N \setminus C \mid w_j \geq w_i, \text{ pour tout } i \in C \} \quad (3.38)$$

La généralisation de (3.36) à l'inégalité de couverture étendue est donnée par la proposition suivante.

Proposition 3.3. [84] Si C est une couverture, alors :

$$\sum_{j \in E(C)} x_j \leq |C| - 1 \quad (3.39)$$

est une inégalité valide pour \mathcal{P} .

Exemple 3.2. Reprenons l'exemple (3.1) [84]. Une inégalité valide obtenue par la proposition (3.39) avec $E(C_4)$ est :

$$\sum_{j=1}^5 x_j \leq 3$$

Cette inégalité domine les inégalités (3.32), (3.33), (3.34) et (3.35), générées par C_1, C_2, C_3 et C_4 , et fournit également une inégalité valide.

Balas(1975) [5], Hammer, Johnson et Peled (1975) [54] et Wolsey (1975) [107] ont donné les conditions nécessaires et suffisantes pour que (3.39) définisse une facette pour le polytope du sac à dos.

3.4.1.2 Inégalité liftée de couverture

Une forme tout à fait générale d'une facette résultante des couvertures minimales est donnée par la proposition suivante.

Proposition 3.4. [84] Si C est une couverture minimale pour \mathbb{P} et (C_1, C_2) est une partition de C avec $C_1 \neq \emptyset$, alors \mathcal{P} a une facette définie par une inégalité de la forme :

$$\sum_{j \in N \setminus C} \alpha_j x_j + \sum_{j \in C_2} \gamma_j x_j + \sum_{j \in C_1} x_j \leq |C_1| - 1 + \sum_{j \in C_2} \gamma_j \quad (3.40)$$

avec $\alpha_j \geq 0$, pour tout $j \in N \setminus C$ et $\gamma_j \geq 0$, pour tout $j \in C_2$.

3.4.1.3 Inégalité liftée simple de couverture

Quand nous appliquons le lifting aux inégalités minimales de couverture, nous obtenons des inégalités de la forme :

$$\sum_{j \in C} x_j + \sum_{j \in N \setminus C} \alpha_j x_j \leq |C| - 1 \quad (3.41)$$

Balas et Zemel ont fourni une procédure de lifting simultané qui calcule les facettes obtenues à partir des couvertures minimales. Les résultats de Balas et de Zemel ont été généralisés par Nemhauser et Vance (1994) [83]. Des résultats au sujet de la complexité informatique pour obtenir des inégalités liftées de couverture peuvent être trouvés dans Hartvigsen et Zemel (1992) [56].

3.4.2 L'inégalité de $(1, k)$ -configurations

Une généralisation de l'inégalité minimale de couverture est donnée par une inégalité, appelée inégalité de $(1, k)$ -configuration, étudiée par Padberg (1980) [87].

Définition 3.4. Un ensemble $N_1 \cup \{\alpha\}$ avec $N_1 \subseteq N$ et $\alpha \in N \setminus N_1$ est appelé une $(1, k)$ -configuration si $\sum_{j \in N_1} w_j \leq c$ et $K \cup \{\alpha\}$ est une couverture minimale pour tout $K \subseteq N_1$ avec $|K| = k$.

Définition 3.5. Pour une $(1, k)$ -configuration donnée, $N_1 \cup \{\alpha\}$ avec $K \subseteq N_1$ et $r = |R| \geq k$, l'inégalité :

$$(r - k + 1)x_\alpha + \sum_{j \in R} x_j \leq r \quad (3.42)$$

est appelée inégalité de $(1, k)$ -configuration correspondant à $N_1 \cup \{\alpha\}$ et $R \subseteq N_1$.

L'inégalité de $(1, k)$ -configuration est une inégalité valide pour \mathcal{P} . Pour $k = |N_1|$, elle est réduite à l'inégalité minimale de couverture. Padberg a démontré que l'ensemble complet des facettes du polytope $\mathcal{P}_{N_1 \cup \{\alpha\}}$ est déterminé par des inégalités de $(1, k)$ -configuration.

3.4.3 L'inégalité de poids

Weismantel [106] a proposé une autre classe d'inégalités appelée inégalités de poids.

Définition 3.6. Supposons que le sous-ensemble $T \subseteq N$ satisfait $\sum_{j \in T} w_j < d$ et définissons la capacité résiduelle comme $r = d - \sum_{j \in T} w_j$. L'inégalité de poids définie sur T est donnée par :

$$\sum_{j \in T} w_j x_j + \sum_{j \in N \setminus T} \max\{0, w_j - r\} x_j \leq \sum_{j \in T} w_j \quad (3.43)$$

Weismantel a montré que l'inégalité de poids est valide pour \mathcal{P} .

Lifting simultané de deux ensembles de variables

L' *inégalité de couverture représente une face pour le polytope du sac à dos et peut être liftée pour définir une facette. Notre objectif est de développer un algorithme qui permet de lifter simultanément deux ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Nous commençons d'abord par la présentation de cet algorithme et nous énonçons les conditions nécessaires pour que les inégalités non-dominées qu'il génère soient des facettes pour le polytope du problème de sac à dos. Nous effectuerons par la suite une comparaison entre cet algorithme et MLSA [98] basée sur le déroulement d'un exemple.*

4.1 Lifting simultané de deux ensembles de variables

Nous allons développer un algorithme qui permet de lifter simultanément deux ensembles de variables E_1 et E_2 en une inégalité de couverture. Les inégalités générées par cet algorithme peuvent avoir les deux formes suivantes :

$$\sum_{i \in C} x_i + \alpha' \sum_{j \in E_1} x_j \leq |C| - 1 \quad (4.1)$$

$$\sum_{i \in C} x_i + \alpha^1 \sum_{j \in E_1} x_j + \alpha^2 \sum_{j \in E_2} x_j \leq |C| - 1 \quad (4.2)$$

Sachant que les trois ensembles C , E_1 et E_2 sont triés selon l'ordre décroissant de leurs coefficients w_j , tel que : $w_i \leq w_{i+1}$, $w_{j^1} \leq w_{j^1+1}$ et $w_{j^2} \leq w_{j^2+1}$ pour C , E_1 et E_2 respectivement.

4.1.1 Principe de l'algorithme

Le principe de l'algorithme consiste à parcourir l'ensemble E_1 et prendre un nombre de variables, noté q_1 , pour les lifter. Puis à chaque itération de ce parcours voir si c'est possible ($T_1 \leq d$) de parcourir le deuxième ensemble E_2 et de prendre un nombre de variables, noté q_2 , et les lifter à leur tour.

Nous choisissons p indices de la couverture minimale correspondant aux plus petits coefficients w_j et nous les fixons à un. La somme de ces coefficients w_j est donnée par C_Σ .

Nous commençons par choisir le premier élément j_1^1 de l'ensemble E_1 et mettre la valeur de E_Σ^1 à $w_{j_1^1}$, $E_\#^1 = 1$, $q_1 = 1$ et $\alpha' = \infty$ puis calculer le premier total $T_1 = C_\Sigma + E_\Sigma^1$:

Étape 1 : Si $T_1 \leq d$, alors p éléments de C et q_1 éléments de $\{j_1^1, \dots, j_{E_\#^1}^1\}$ ne forment pas une couverture, donc, il existe le point faisable correspondant. Cependant la valeur du coefficient α' doit être changée de sorte que ce point avec p et q_1 éléments satisfont l'inégalité (4.1) à égalité. Les valeurs de q_1 et $E_\#^1$ sont incrémentées de 1.

Nous choisissons le premier élément j_1^2 de l'ensemble E_2 et nous mettons la valeur de E_Σ^2 à $w_{j_1^2}$, $E_\#^2 = 1$, $q_2 = 1$, $\alpha^1 = \alpha'$ et $\alpha^2 = \alpha'$. Puis, nous calculons le deuxième totale $T_2 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2$.

Étape 1.1 : Si $T_2 \leq d$, alors, p éléments de C , q_1 éléments de $\{j_1^1, \dots, j_{E_\#^1}^1\}$ et q_2 éléments de $\{j_1^2, \dots, j_{E_\#^2}^2\}$ ne forment pas une couverture, donc, il existe le point faisable correspondant, d'où, les valeurs des coefficients α^1 et α^2 doivent être changées de sorte que ce point

avec p , q_1 et q_2 éléments satisfont l'inégalité (4.2) à égalité. Les valeurs de q_2 et $E_{\#}^2$ sont incrémentées de 1.

Étape 1.2 : Sinon, si $T_2 > d$, les p éléments de C , q_1 éléments de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$ et q_2 éléments de $\{j_1^2, \dots, j_{E_{\#}^2}^2\}$ forment une couverture, donc, les coefficients du lifting α^1 et α^2 restent inchangés ($\alpha_{E_{\#}^1, E_{\#}^2}^1 = \alpha_{E_{\#}^1, E_{\#}^2 - 1}^1$, $\alpha_{E_{\#}^1, E_{\#}^2}^2 = \alpha_{E_{\#}^1, E_{\#}^2 - 1}^2$). Maintenant, uniquement $E_{\#}^2$ sera incrémenté de un.

Cette alternative sera répété jusqu'à ce que $E_{\#}^2$ soit égal à $|E_2|$.

Étape 2 : Sinon, si $T_1 > d$, les p éléments de C et q_1 éléments de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$ forment une couverture et donc le coefficient du lifting α' reste inchangé ($\alpha'_{p, E_{\#}^1} = \alpha'_{p, E_{\#}^1 - 1}$) et uniquement $E_{\#}^1$ sera incrémenté de un.

Cette deuxième alternative sera répété jusqu'à ce que $E_{\#}^1$ soit égal à $|E_1|$.

Le processus en entier sera répété pour tout $p \in \{|C| - 1, \dots, 0\}$. Lorsque nous terminons de dérouler l'algorithme, nous maintenons les valeurs de $\alpha'_{p, E_{\#}^1}$, $\alpha_{E_{\#}^1, E_{\#}^2}^{1*}$ et $\alpha_{E_{\#}^1, E_{\#}^2}^{2*}$ qui correspondent aux $\text{minimum}_{p \in \{0, 1, \dots, |C| - 1\}} \{\alpha'_{p, E_{\#}^1}\}$, $\text{minimum}_{p \in \{0, 1, \dots, |C| - 1\}} \{\alpha_{E_{\#}^1, E_{\#}^2}^{1*}\}$ et $\text{minimum}_{p \in \{0, 1, \dots, |C| - 1\}} \{\alpha_{E_{\#}^1, E_{\#}^2}^{2*}\}$, ces derniers représentent les coefficients du lifting simultanés optimaux de $E_{\#}^1$ et $E_{\#}^1 + E_{\#}^2$ variables respectivement. Cependant, nous pouvons facilement générer l'inégalité maximale du lifting simultané en choisissant les inégalités qui ont la plus grande valeur de $E_{\#}^1$ et $E_{\#}^2$ pour $\alpha'_{p, E_{\#}^1}$, $\alpha_{E_{\#}^1, E_{\#}^2}^{1*}$ et $\alpha_{E_{\#}^1, E_{\#}^2}^{2*}$ donnés respectivement.

4.1.2 Calcul des coefficients α^1 et α^2

Pour les inégalités de la forme (4.1), le coefficient α' se calcule comme suit :

$$\alpha' = \frac{|C| - 1 - p}{q_1} \quad (4.3)$$

Le calcul des coefficients α^1 et α^2 , pour les inégalités de la forme (4.2), se fait de la manière suivante :

Si $T_2 \leq d$, cela implique que $C_{\Sigma} + E_{\Sigma}^1 \leq d$ (resp : $C_{\Sigma} + E_{\Sigma}^2 \leq d$), d'où, p éléments de C et q_1 (resp : q_2) éléments de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$ (resp : $\{j_1^2, \dots, j_{E_{\#}^2}^2\}$) ne forment pas une couverture, donc, il existe le point faisable correspondant et la valeur du coefficient α^1 (resp : α^2) doit être changée de sorte que ce point avec p et q_1 (resp : q_2) éléments satisfont l'inégalité (4.2) à égalité. Donc, nous avons :

$$\alpha^1 = \frac{|C| - 1 - p}{q_1} \quad (4.4)$$

$$\alpha^2 = \frac{|C| - 1 - p}{q_2} \quad (4.5)$$

Par substitution dans l'inégalité (4.2), pour $\sum_{i \in C} x_i = p$, $\sum_{j \in E_1} x_j = q_1$ et $\sum_{j \in E_2} x_j = q_2$, nous aurons :

$$\frac{|C| - 1 - p}{q_1} \times q_1 + \frac{|C| - 1 - p}{q_2} \times q_2 = 2(|C| - 1 - p) > |C| - 1 - p$$

En multipliant les coefficients α^1 et α^2 par λ_1 et λ_2 respectivement, nous aurons :

$$\alpha^1 = \frac{|C| - 1 - p}{q_1} \times \lambda_1 \quad (4.6)$$

$$\alpha^2 = \frac{|C| - 1 - p}{q_2} \times \lambda_2 \quad (4.7)$$

où $\lambda_1, \lambda_2 \in]0, 1[$. Par substitution dans l'inégalité (4.2) :

$$\frac{|C| - 1 - p}{q_1} \times \lambda_1 \times q_1 + \frac{|C| - 1 - p}{q_2} \times \lambda_2 \times q_2 = (\lambda_1 + \lambda_2)(|C| - 1 - p)$$

Pour que l'inégalité (4.2) soit satisfaite à égalité nous devons avoir :

$$\lambda_1 + \lambda_2 = 1 \quad (4.8)$$

Nous avons choisi de prendre :

- $\lambda_1 = \frac{q_1}{q_1 + q_2}$: Le taux des éléments de E_1 pris dans $N \setminus C$;
- $\lambda_2 = \frac{q_2}{q_1 + q_2}$: Le taux des éléments de E_2 pris dans $N \setminus C$.

D'où :

$$\left. \begin{aligned} \alpha^1 &= \frac{|C| - 1 - p}{q_1} \times \frac{q_1}{q_1 + q_2} = \frac{|C| - 1 - p}{q_1 + q_2} \\ \alpha^2 &= \frac{|C| - 1 - p}{q_2} \times \frac{q_2}{q_1 + q_2} = \frac{|C| - 1 - p}{q_1 + q_2} \end{aligned} \right\} \Rightarrow \alpha^1 = \alpha^2 = \frac{|C| - 1 - p}{q_1 + q_2} \quad (4.9)$$

4.1.3 Formes des inégalités générées

Puisque $\alpha^1 = \alpha^2$, alors les inégalités générées par l'algorithme 2 ont la forme (4.1) ou bien la forme suivante :

$$\sum_{i \in C} x_i + \alpha \left(\sum_{j \in E_1} x_j + \sum_{j \in E_2} x_j \right) \leq |C| - 1 \quad (4.10)$$

où $\alpha = \alpha^1 = \alpha^2$

Algorithme 2 : Algorithme pour le lifting simultané de deux ensembles de variables

Données : Une contrainte du problème du sac à dos : $\sum_{j \in N} w_j x_j \leq d$.

Les ensembles de variables C , E_1 et E_2 ($N \setminus C = E_1 \cup E_2$).

Résultats : Des inégalités de la forme (4.1) et (4.10)

$p := |C| - 1$;

while $p \geq 0$ **do**

$$C_\Sigma := \sum_{r=|C|-p+1}^{|C|} w_{i_r} ; \quad \alpha'_{p,0} := \infty ; \quad q_1 := 1 ; \quad E_\#^1 := 1 ;$$

while $E_\#^1 \leq |E_1|$ **do**

$$E_\Sigma^1 := \sum_{r=E_\#^1-q_1+1}^{E_\#^1} w_{j_r} ; \quad T_1 = C_\Sigma + E_\Sigma^1 ;$$

if $T_1 \leq b$ **then**

$$\alpha'_{p,E_\#^1} := \frac{|C|-1-p}{q_1} ; \quad q_1 := q_1 + 1 ; \quad \alpha_{E_\#^1,0} := \alpha'_{p,E_\#^1} ;$$

$$q_2 := 1 ; \quad E_\#^2 := 1 ;$$

while $E_\#^2 \leq |E_2|$ **do**

$$E_\Sigma^2 := \sum_{r=E_\#^2-q_2+1}^{E_\#^2} w_{j_r} ; \quad T_2 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2 ;$$

if $T_2 \leq b$ **then**

$$\alpha_{E_\#^1,E_\#^2} := \frac{|C|-1-p}{q_1+q_2} ; \quad q_2 := q_2 + 1 ;$$

else

$$\alpha_{E_\#^1,E_\#^2} := \alpha_{E_\#^1,E_\#^2-1} ;$$

$$E_\#^2 := E_\#^2 + 1 ;$$

else

$$\alpha'_{p,E_\#^1} := \alpha'_{p,E_\#^1-1} ;$$

$$E_\#^1 := E_\#^1 + 1 ;$$

$p := p - 1$;

Exemple 4.1. *Considérons la contrainte du sac à dos suivante :*

$$29x_1 + 29x_2 + 29x_3 + 28x_4 + 14x_5 + 14x_6 + 12x_7 + 5x_8 + 1x_9 + 1x_{10} + 1x_{11} \leq 90. \quad (4.11)$$

Une couverture pour ce problème de sac à dos est $C = \{1, 2, 3, 4\}$, parce que la somme des coefficients $29 + 29 + 29 + 28 = 115$ est plus grande que le second membre de l'inégalité (4.11) (90). C'est

également une couverture minimale, car si n'importe quel indice est enlevé de la couverture, celle-ci n'est plus une couverture. Clairement, l'inégalité valide de couverture est $x_1 + x_2 + x_3 + x_4 \leq 3$. Commençons par la couverture précédente, nous liftons simultanément toutes les autres variables de l'ensemble $N \setminus C$, mais avant, nous allons d'abord partitionner cet ensemble en deux sous ensembles E_1 et E_2 avec $E_1 = \{5, 6, 7, 8\}$ et $E_2 = \{9, 10, 11\}$.

Nous commençons les itérations : nous fixons p à $|C| - 1 = 3$ et C_Σ est donné par la somme des plus petits w_j , $C_\Sigma = w_2 + w_3 + w_4 = 29 + 29 + 28 = 86$ et le coefficient de lifting $\alpha'_{3,0}$ est initialisé à ∞ . Dans l'étape principale, nous choisissons le premier indice dans l'ensemble E_1 i.e. 5, fixer $q_1 = 1$, $E_\#^1 = 1$, la somme des coefficients de E_1 est indiquée par $E_\Sigma^1 = w_5 = 14$ et $T_1 = C_\Sigma + E_\Sigma^1 = 86 + 14 = 100$. Puisque le total est plus grand que le second membre de l'inégalité (4.11) (90), $\alpha'_{3,1} = \alpha'_{3,0} = \infty$. Dans la prochaine itération, q_1 demeure sans changement, c-à-d, $q_1 = 1$, $E_\#^1 = 2$, $E_\Sigma^1 = w_6 = 14$ et $T_1 = C_\Sigma + E_\Sigma^1 = 86 + 14 = 100$. Encore, le total est plus grand que c , ainsi, $\alpha'_{3,2} = \infty$. Incrémentons $E_\#^1 = 3$, $q_1 = 1$, $E_\Sigma^1 = w_7 = 12$ et $T_1 = 86 + 12 = 98$ qui est plus grand que 90, donc, $\alpha'_{3,3} = \infty$. Dans la prochaine itération, $E_\#^1 = 4$, $q_1 = 1$ et $E_\Sigma^1 = w_8 = 5$, donc, $T_1 = 86 + 5 = 91$ et encore $\alpha'_{3,4} = \infty$. Par conséquent, quand $p = 3$, aucune variable peut être liftée de E_1 et $\alpha'_{3,\cdot} = \infty$.

Passons à la prochaine itération, nous commençons par décrémenter le p . Maintenant, $p = 2$, $C_\Sigma = w_3 + w_4 = 57$, $q_1 = 1$, $E_\#^1 = 1$, $E_\Sigma^1 = w_5 = 14$ et $T_1 = C_\Sigma + E_\Sigma^1 = 57 + 14 = 71$. Puisque 71 est inférieur à 90, le coefficient de lifting est $\alpha'_{2,1} = \frac{3-2}{1} = 1$. L'inégalité proposée avec $E_\#^1$ variables liftées peut être écrite comme suit :

$$x_1 + x_2 + x_3 + x_4 + 1x_5 \leq 3$$

Puisque $T_1 = 71 < 90$, alors nous pouvons lifter des éléments de l'ensemble E_2 . Nous initialisons $\alpha_{1,0} = \alpha'_{2,1} = 1$, puis, nous commençons à itérer la sous étape principale en choisissant le premier indice dans l'ensemble E_2 i.e. 9, fixer $q_2 = 1$ et $E_\#^2 = 1$, la somme des coefficients de E_2 est indiquée par $E_\Sigma^2 = w_9 = 1$ et $T_2 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2 = 71 + 1 = 72$. Puisque 73 est inférieur à 90, le coefficient du lifting est $\alpha_{1,1} = \frac{3-2}{1+1} = \frac{1}{2}$. L'inégalité proposée avec $E_\#^1 + E_\#^2$ variables liftées peut être écrite comme suit :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_9) \leq 3$$

Puisque le total T_2 est inférieur à 90, une variable en plus peut être additionnée dans l'ensemble $\{j_1^2, \dots, j_{E_\#^2}^2\}$ et donc, q_2 et $E_\#^2$ sont incrémentés. Dans la deuxième itération, $q_2 = 2$, $E_\#^2 = 2$, $E_\Sigma^2 = w_9 + w_{10} = 1 + 1 = 2$ et $T_2 = 71 + 2 = 73$. Encore, 73 est inférieur à 90, donc, le coefficient du lifting

$\alpha_{1,2} = \frac{3-2}{1+2} = \frac{1}{3}$ et l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_9 + x_{10}) \leq 3.$$

Ensuite, q_2 et $E_{\#}^2$ sont incrémentés à $q_2 = 3$, $E_{\#}^2 = 3$, $E_{\Sigma}^2 = 1 + 1 + 1 = 3$ et $T_2 = 71 + 3 = 74$. Le total est toujours inférieur à 90, d'où, $\alpha_{1,3} = \frac{3-2}{1+3} = \frac{1}{4}$, et ainsi, l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{4}(x_5 + x_9 + x_{10} + x_{11}) \leq 3.$$

Nous retournons à l'étape principale, puisque, le total T_1 est inférieur à 90, une variable en plus peut être additionnée dans l'ensemble $E_{\#}^1$, donc, q_1 et $\{j_1^2, \dots, j_{E_{\#}^1}^2\}$ sont incrémentés : $q_1 = 2$, $E_{\#}^1 = 2$, $E_{\Sigma}^1 = w_5 + w_6 = 14 + 14 = 28$ et $T_1 = 57 + 28 = 85$. Encore, 85 est inférieur à 90, donc, le coefficient du lifting $\alpha'_{2,2} = \frac{3-2}{2} = \frac{1}{2}$ et l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6) \leq 3.$$

Nous avons $T_1 = 85 < 90$, donc, nous pouvons lifter des éléments de l'ensemble E_2 . Nous initialisons $\alpha_{2,0} = \alpha'_{2,2}$ puis nous commençons à itérer la sous étape principale en choisissant le premier indice dans l'ensemble E_2 i.e. 9, fixer $q_2 = 1$ et $E_{\#}^2 = 1$, la somme des coefficients de E_2 est indiquée par $E_{\Sigma}^2 = w_9 = 1$ et $T_2 = C_{\Sigma} + E_{\Sigma}^1 + E_{\Sigma}^2 = 85 + 1 = 86$. Puisque 86 est inférieur à 90, le coefficient du lifting est $\alpha_{2,1} = \frac{3-2}{2+1} = \frac{1}{3}$. L'inégalité proposée avec $E_{\#}^1 + E_{\#}^2$ variables liftées peut être écrite comme suit :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_9) \leq 3$$

Nous avons T_2 est inférieur à 90, une variable en plus peut être additionnée dans l'ensemble $E_{\#}^2$, donc, q_2 et $E_{\#}^2$ sont incrémentés. Dans la deuxième itération, $q_2 = 2$, $E_{\#}^2 = 2$, $E_{\Sigma}^2 = w_9 + w_{10} = 1 + 1 = 2$ et $T_2 = 85 + 2 = 87$. Encore, 87 est inférieur à 90, donc, le coefficient du lifting $\alpha_{2,2} = \frac{3-2}{2+2} = \frac{1}{4}$ et l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{4}(x_5 + x_6 + x_9 + x_{10}) \leq 3.$$

Nous incrémentons q_2 et $E_{\#}^2$ à $q_2 = 3$, $E_{\#}^2 = 3$, $E_{\Sigma}^2 = 1 + 1 + 1 = 3$ et $T_2 = 85 + 3 = 88$. Le total est inférieur à 90, d'où, $\alpha_{2,3} = \frac{3-2}{2+3} = \frac{1}{5}$ et ainsi, l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{5}(x_5 + x_6 + x_9 + x_{10} + x_{11}) \leq 3.$$

Retournons à l'étape principale, nous avons le total T_1 est inférieur à 90, donc, une variable en plus peut être additionnée dans l'ensemble $\{j_1^1, \dots, j_{E_\#^1}^1\}$, donc, q_1 et $E_\#^1$ sont incrémentés $q_1 = 3$, $E_\#^1 = 3$, $E_\Sigma^1 = w_5 + w_6 + w_7 = 14 + 14 + 12 = 40$ et $T_1 = 57 + 40 = 97$. Ici, le total est plus grand que 90, par conséquent, $\alpha'_{2,3} = \alpha'_{2,2} = \frac{1}{2}$, ainsi, l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$$

Maintenant, $E_\#^1$ s'incrémente à 4 et q_1 reste égal à 3. Ceci indique que trois des quatre indices peuvent être choisis. Donc, E_Σ^1 est maintenant égal à $w_6 + w_7 + w_8 = 31$, $C_\Sigma = 57$ et $T_1 = C_\Sigma + E_\Sigma^1 = 88$ qui est inférieur à 90. Ainsi, le nouveau coefficient du lifting est $\alpha'_{2,4} = \frac{3-2}{3} = \frac{1}{3}$, par conséquent, l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7 + x_8) \leq 3,$$

Le total $T_1 = 88 < 90$, donc, nous pouvons lifter des éléments de l'ensemble E_2 . Nous initialisons $\alpha_{4,0} = \alpha'_{2,4} = \frac{1}{3}$, puis, nous commençons par itérer la sous étape principale en choisissant le premier indice dans l'ensemble E_2 i.e. 9, puis fixer $q_2 = 1$ et $E_\#^2 = 1$, la somme des coefficients de E_2 est indiquée par $E_\Sigma^2 = w_9 = 1$ et $T_2 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2 = 88 + 1 = 89$, d'où, le coefficient du lifting est $\alpha_{4,1} = \frac{3-2}{3+1} = \frac{1}{4}$. L'inégalité proposée avec $E_\#^1 + E_\#^2$ variables liftées peut être écrite comme suit :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{4}(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$$

Nous avons T_2 est égal à 89, une variable en plus peut être additionnée dans l'ensemble $E_\#^2$, donc, q_2 et $E_\#^2$ sont incrémentés $q_2 = 2$, $E_\#^2 = 2$, $E_\Sigma^2 = w_9 + w_{10} = 1 + 1 = 2$ et $T_2 = 88 + 2 = 90$. Ici, le total est égal à 90, par conséquent, le coefficient du lifting $\alpha_{4,2} = \frac{3-2}{3+2} = \frac{1}{5}$ et l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{5}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 3.$$

Nous incrémentons q_2 et $E_\#^2$ à $q_2 = 3$, $E_\#^2 = 3$, $E_\Sigma^2 = 1 + 1 + 1 = 3$ et $T_2 = 88 + 3 = 91$. Le total est supérieur à 90, d'où, $\alpha_{4,3} = \alpha_{4,2} = \frac{1}{5}$, ainsi, l'inégalité proposée est :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{5}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3.$$

Pour ne pas alourdir l'exemple, nous allons résumer les résultats du déroulement dans les tableaux 4.1, 4.2 et 4.3. Les valeurs de $\alpha'_{p,E_\#^1}$ et $\alpha_{E_\#^1,E_\#^2}$ générées par la l'algorithme 2 sont résumées dans les tableaux 4.4 et 4.5.

$E_{\#}^1$	$E_{\#}^2$	q_2	E_{Σ}^2	T_2	α	inégalité $_{E_1}$	inégalité $_{E_1 \cup E_2}$
$q_1 = 1$	$\alpha'_{2,1} = 1$	1	1	72	$\frac{1}{2}$		$\sum_{i=1}^4 x_i + (1/2)(x_5 + 1x_9) \leq 3$
w_5	$E_{\Sigma}^1 = 14$	2	2	73	$\frac{1}{3}$	$\sum_{i=1}^4 x_i + 1x_5 \leq 3$	$\sum_{i=1}^4 x_i + (1/3)(x_5 + x_9 + x_{10}) \leq 3$
	$T_1 = 71$	3	3	74	$\frac{1}{4}$		$\sum_{i=1}^4 x_i + (1/4)(x_5 + x_9 + x_{10} + x_{11}) \leq 3$
$q_1 = 2$	$\alpha'_{2,2} = \frac{1}{2}$	1	1	86	$\frac{1}{3}$		$\sum_{i=1}^4 x_i + (1/3)(x_5 + x_6 + x_9) \leq 3$
w_5, w_6	$E_{\Sigma}^1 = 28$	2	2	87	$\frac{1}{4}$	$\sum_{i=1}^4 x_i + \frac{1}{2}(x_5 + x_6) \leq 3$	$\sum_{i=1}^4 x_i + (1/4)(x_5 + x_6 + x_9 + x_{10}) \leq 3$
	$T_1 = 85$	3	3	88	$\frac{1}{5}$		$\sum_{i=1}^4 x_i + (1/5)(x_5 + x_6 + x_9 + x_{10} + x_{11}) \leq 3$
$q_1 = 3$	$\alpha'_{2,3} = \frac{1}{2}$	-	-	-	-		---
w_5, w_6	$E_{\Sigma}^1 = 40$	-	-	-	-	$\sum_{i=1}^4 x_i + \frac{1}{2} \sum_{i=5}^7 x_i \leq 3$	---
w_7	$T_1 = 97$	-	-	-	-		---
$q_1 = 3$	$\alpha'_{2,4} = \frac{1}{3}$	1	1	89	$\frac{1}{4}$		$\sum_{i=1}^4 x_i + (1/4)(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$
w_6, w_7	$E_{\Sigma}^1 = 31$	2	2	90	$\frac{1}{5}$	$\sum_{i=1}^4 x_i + \frac{1}{3} \sum_{i=5}^8 x_i \leq 3$	$\sum_{i=1}^4 x_i + (1/5)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 3$
w_8	$T_1 = 88$	3	3	91	$\frac{1}{5}$		$\sum_{i=1}^4 x_i + (1/5)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$

TAB. 4.1: Les résultats du déroulement de l'exemple 4.1 pour $p = 2$ et $C_{\Sigma} = 57$

	$E_1^\#$	$E_2^\#$	q_2	$E_2^\#$	T_2	α	Inégalité E_1	Inégalité $E_1 \cup E_2$
$q_1 = 1$	$\alpha'_{1,1} = 2$	1	1	1	43	1	$\sum_{i=1}^4 x_i + 2x_5 \leq 3$	$\sum_{i=1}^4 x_i + 1x_5 + 1x_9 \leq 3$
		2	2	2	44	$\frac{2}{3}$		$\sum_{i=1}^4 x_i + (2/3)(x_5 + x_9 + x_{10}) \leq 3$
w_5	$E_2^1 = 14$	2	2	2	44	$\frac{2}{3}$	$\sum_{i=1}^4 x_i + 2x_5 \leq 3$	$\sum_{i=1}^4 x_i + (2/3)(x_5 + x_9 + x_{10}) \leq 3$
		3	3	3	45	$\frac{1}{2}$		$\sum_{i=1}^4 x_i + (1/2)(x_5 + x_9 + x_{10} + x_{11}) \leq 3$
$q_1 = 2$	$\alpha'_{1,2} = 1$	1	1	1	57	$\frac{2}{3}$	$\sum_{i=1}^4 x_i + 1x_5 + 1x_6 \leq 3$	$\sum_{i=1}^4 x_i + (2/3)(x_5 + x_6 + x_9) \leq 3$
		2	2	2	58	$\frac{1}{2}$		$\sum_{i=1}^4 x_i + (1/2)(x_5 + x_6 + x_9 + x_{10}) \leq 3$
w_5, w_6	$E_2^1 = 28$	2	2	2	58	$\frac{1}{2}$	$\sum_{i=1}^4 x_i + 1x_5 + 1x_6 \leq 3$	$\sum_{i=1}^4 x_i + (2/5)(x_5 + x_6 + x_9 + x_{10} + x_{11}) \leq 3$
		3	3	3	59	$\frac{2}{5}$		$\sum_{i=1}^4 x_i + (2/5)(x_5 + x_6 + x_9 + x_{10} + x_{11}) \leq 3$
$q_1 = 3$	$\alpha'_{1,3} = \frac{2}{3}$	1	1	1	69	$\frac{1}{2}$	$\sum_{i=1}^4 x_i + \frac{2}{3} \sum_{i=5}^7 x_i \leq 3$	$\sum_{i=1}^4 x_i + (1/2)(x_5 + x_6 + x_7 + x_9) \leq 3$
		2	2	2	70	$\frac{2}{5}$		$\sum_{i=1}^4 x_i + (2/5)(x_5 + x_6 + x_7 + x_9 + x_{10}) \leq 3$
w_5, w_6	$E_2^1 = 40$	2	2	2	70	$\frac{2}{5}$	$\sum_{i=1}^4 x_i + \frac{2}{3} \sum_{i=5}^7 x_i \leq 3$	$\sum_{i=1}^4 x_i + (2/5)(x_5 + x_6 + x_7 + x_9 + x_{10}) \leq 3$
		3	3	3	71	$\frac{1}{3}$		$\sum_{i=1}^4 x_i + (1/3)(x_5 + x_6 + x_7 + x_9 + x_{10} + x_{11}) \leq 3$
w_7	$T_1 = 68$	3	3	3	71	$\frac{1}{3}$	$\sum_{i=1}^4 x_i + \frac{1}{2} \sum_{i=5}^8 x_i \leq 3$	$\sum_{i=1}^4 x_i + (2/5)(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$
		1	1	1	74	$\frac{2}{5}$		$\sum_{i=1}^4 x_i + (1/3)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 3$
$q_1 = 4$	$\alpha'_{1,3} = \frac{1}{2}$	1	1	1	74	$\frac{2}{5}$	$\sum_{i=1}^4 x_i + \frac{1}{2} \sum_{i=5}^8 x_i \leq 3$	$\sum_{i=1}^4 x_i + (1/3)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 3$
		2	2	2	75	$\frac{1}{3}$		$\sum_{i=1}^4 x_i + (2/7)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$
w_5, w_6	$E_2^1 = 45$	2	2	2	75	$\frac{1}{3}$	$\sum_{i=1}^4 x_i + \frac{1}{2} \sum_{i=5}^8 x_i \leq 3$	$\sum_{i=1}^4 x_i + (2/7)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$
		3	3	3	76	$\frac{2}{7}$		$\sum_{i=1}^4 x_i + (2/7)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$
w_7, w_8	$T_1 = 73$	3	3	3	76	$\frac{2}{7}$		

TAB. 4.2: Les résultats du déroulement de l'exemple 4.1 pour $p = 1$ et $G_2 = 28$

$E_{\#}^1$	$E_{\#}^2$	q_2	F_{Σ}^2	T_2	α	inégalité $_{E_1}$	inégalité $_{E_1 \cup E_2}$
$q_1 = 1$	$\alpha'_{1,1} = 3$	1	1	15	$\frac{3}{2}$		$\sum_{i=1}^4 x_i + \frac{3}{2}(x_5 + x_9) \leq 3$
w_5	$E_{\Sigma}^1 = 14$	2	2	16	1	$\sum_{i=1}^4 x_i + 3x_5 \leq 3$	$\sum_{i=1}^4 x_i + (1)(x_5 + x_9 + x_{10}) \leq 3$
		3	3	17	$\frac{3}{4}$		$\sum_{i=1}^4 x_i + (3/4)(x_5 + x_9 + x_{10} + x_{11}) \leq 3$
		1	1	29	1		$\sum_{i=1}^4 x_i + (1)(x_5 + x_6 + x_9) \leq 3$
$q_1 = 2$	$\alpha'_{1,2} = \frac{3}{2}$	2	2	30	$\frac{3}{4}$	$\sum_{i=1}^4 x_i + \frac{3}{2}(x_5 + x_6) \leq 3$	$\sum_{i=1}^4 x_i + (3/4)(x_5 + x_6 + x_9 + x_{10}) \leq 3$
		3	3	31	$\frac{3}{5}$		$\sum_{i=1}^4 x_i + (3/5)(x_5 + x_6 + x_9 + x_{10} + x_{11}) \leq 3$
		1	1	41	$\frac{1}{2}$		$\sum_{i=1}^4 x_i + (3/4)(x_5 + x_6 + x_7 + x_9) \leq 3$
w_5, w_6	$E_{\Sigma}^1 = 28$	2	2	42	$\frac{2}{5}$	$\sum_{i=1}^4 x_i + \sum_{i=5}^7 x_i \leq 3$	$\sum_{i=1}^4 x_i + (3/5)(x_5 + x_6 + x_7 + x_9 + x_{10}) \leq 3$
		3	3	43	$\frac{1}{3}$		$\sum_{i=1}^4 x_i + (1/2)(x_5 + x_6 + x_7 + x_9 + x_{10} + x_{11}) \leq 3$
		1	1	46	$\frac{3}{4}$		$\sum_{i=1}^4 x_i + (3/4)(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$
$q_1 = 3$	$\alpha'_{1,3} = \frac{4}{3}$	2	2	47	$\frac{3}{5}$	$\sum_{i=1}^4 x_i + \frac{3}{4} \sum_{i=5}^8 x_i \leq 3$	$\sum_{i=1}^4 x_i + (3/5)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 3$
		3	3	48	$\frac{1}{2}$		$\sum_{i=1}^4 x_i + (1/2)(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$
		1	1				
w_5, w_6	$E_{\Sigma}^1 = 45$	2	2				
w_7, w_8	$T_1 = 45$	3	3				

TAB. 4.3: Les résultats du déroulement de l'exemple 4.1 pour $p = 0$ et $C_{\Sigma} = 0$

	$p = 3$		$p = 2$		$p = 1$		$p = 0$		
$E_{\#}^1$	q_1	$\alpha'_{3,E_{\#}^1}$	q_1	$\alpha'_{2,E_{\#}^1}$	q_1	$\alpha'_{1,E_{\#}^1}$	q_1	$\alpha'_{0,E_{\#}^1}$	$\min \alpha_{p,E_{\#}^1}^*$
1	1	∞	1	1	1	2	1	3	1
2	1	∞	2	$\frac{1}{2}$	2	1	2	$\frac{3}{2}$	$\frac{1}{2}$
3	1	∞	3	$\frac{1}{2}$	3	$\frac{2}{3}$	3	1	$\frac{1}{2}$
4	1	∞	3	$\frac{1}{2}$	4	$\frac{1}{2}$	4	$\frac{3}{4}$	$\frac{1}{3}$

TAB. 4.4: Les valeurs de $\alpha'_{p,E_{\#}^1}$ générées par l'algorithme 2

p	$p = 2$				$p = 1$				$p = 0$				$\min \alpha_{E_{\#}^1, E_{\#}^2}^*$
$E_{\#}^2$	1	2	3	4	1	2	3	4	1	2	3	4	
1	$\frac{1}{2}$	$\frac{1}{3}$	-	$\frac{1}{4}$	1	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{3}{2}$	1	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{4}$
2	$\frac{1}{3}$	$\frac{1}{4}$	-	$\frac{1}{5}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{3}$	1	$\frac{3}{4}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{1}{5}$
3	$\frac{1}{4}$	$\frac{1}{5}$	-	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{3}$	$\frac{2}{7}$	$\frac{3}{4}$	$\frac{3}{5}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{5}$

TAB. 4.5: Les valeurs de $\alpha_{E_{\#}^1, E_{\#}^2}$ générées par l'algorithme 2

L'algorithme 2 génère des inégalités dont plusieurs sont dominées par d'autres inégalités. Par exemple, l'inégalité $x_1 + x_2 + x_3 + x_4 + \frac{1}{5}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 3$ est dominée par l'inégalité $x_1 + x_2 + x_3 + x_4 + \frac{1}{5}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$. Ainsi, les inégalités non-dominées sont les inégalités ayant le plus grand nombre de variables pour des valeurs particulières de α et α' . Dans cet exemple, il y a trois inégalités non-dominées :

$$x_1 + x_2 + x_3 + x_4 + 1(x_5) \leq 3 \tag{4.12}$$

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3 \tag{4.13}$$

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{5}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3 \tag{4.14}$$

- L'inégalité (4.12) correspond au minimum des $\alpha'_{p,E_{\#}^1} = 1$ dans le tableau 4.4, pour $E_{\#}^1 = 1$, $p = 2$ et $q_1 = 1$.
- L'inégalité (4.13) correspond au minimum des $\alpha'_{p,E_{\#}^1} = \frac{1}{2}$ dans le tableau 4.4, pour $E_{\#}^1 = 3$, $p = 2$ et $q_1 = 2$.
- L'inégalité (4.14) correspond au minimum des $\alpha_{E_{\#}^1, E_{\#}^2} = \frac{1}{5}$ dans le tableau 4.5, pour $E_{\#}^1 = 4$, $E_{\#}^2 = 3$, $p = 2$, $q_1 = 3$ et $q_2 = 2$.

De l'exemple ci-dessus, il est clair que l'algorithme 2 génère des inégalités liftées maximales fortes.

Les inégalités (4.12), (4.13) et (4.14) représentent des facettes, car nous pouvons exhiber 11 points affinement indépendants pour chacune d'elles qui seront représentés par les matrices A_1 , A_2 et A_3 respectivement. Chaque point est représenté par une colonne. Pour les trois matrices, les quatre premiers points affinement indépendants sont obtenus à partir de la couverture minimale $\{1, 2, 3, 4\}$, les autres points sont obtenus de la manière suivante :

- ◇ Dans la première inégalité (4.12), nous avons $E_{\#}^1 = 1$, $q_1 = 1$, $p = 2$ et $\alpha'_{2,1} = 1$. Le cinquième point $(0, 0, 1, 1, 1, 0, 0, 0, 0, 0)$ est obtenu en choisissant les deux indices les plus grands dans C et un indice de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$, à savoir ; j_1^1 . Les six autres points sont obtenus en choisissant les deux indices les plus grands dans C et un indice de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$, à savoir ; j_1^1 , et un autre indice de ce qui reste des indices $(N \setminus (C \cup \{j_1^1\}))$. Ces points sont affinement indépendants.

$$A_1 = \left(\begin{array}{cccc|c} 0 & 1 & 1 & 1 & \mathbf{0} \\ 1 & 0 & 1 & 1 & \mathbf{1} \\ 1 & 1 & 0 & 1 & \\ 1 & 1 & 1 & 0 & \\ 0 & 0 & 0 & 0 & \\ \hline & \mathbf{0} & & & \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \\ & & & & \mathbf{I} \end{array} \right)$$

TAB. 4.6: Les points affinement indépendants pour $E_{\#}^1 = 1$, $q_1 = 1$, $p = 2$ et $\alpha'_{2,1} = 1$

- ◇ Pour la deuxième inégalité (4.13), $E_{\#}^1 = 3$, $q_1 = 2$, $p = 2$ et $\alpha'_{2,3} = \frac{1}{2}$. Trois points sont obtenus par une permutation cyclique de la combinaison $p = 2$, $q_1 = 2$ et $E_{\#}^1 = 3$ qui veut dire que deux indices parmi trois indices de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$ sont sélectionnés. Les quatre derniers points sont obtenus en choisissant les deux indices les plus grands dans C , deux indices les plus grand dans $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$ et un autre indice de ce qui reste des indices $(N \setminus (C \cup \{j_1^1, \dots, j_{E_{\#}^1}^1\}))$. Ces points sont affinement indépendants.
- ◇ La troisième inégalité (4.14), $E_{\#}^1 = 4$, $E_{\#}^2 = 3$, $p = 2$, $q_1 = 3$, $q_2 = 2$ et $\alpha_{4,3} = \frac{1}{5}$. Trois points sont obtenus en choisissant les deux plus grands indices dans C , les trois plus grands indices dans E_2 et deux indices dans E_1 obtenus par une permutation cyclique parmi les trois plus grands indices dans $E_1(\{j_{E_{\#}^1 - q_1 + 1}^1, \dots, j_{E_{\#}^1}^1\})$. Le prochain point est obtenu en choisissant les deux plus grand indices dans C et les trois plus grand indices dans E_2 et le premier et le dernier élément dans E_1 . Les trois points qui reste sont obtenus en choisissant les deux plus grand indices dans C et les trois plus grand indices dans E_1 et deux indices de E_2 obtenus par une permutation cyclique parmi les trois plus grand indices dans $E_2(\{j_{E_{\#}^2 - q_2 + 1}^2, \dots, j_{E_{\#}^2}^2\})$. Ces points sont affinement indépendants.

$$A_2 = \left(\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & & & \\ 1 & 0 & 1 & 1 & & & \\ 1 & 1 & 0 & 1 & & & \\ 1 & 1 & 1 & 0 & & & \\ \hline & & & & 0 & & \\ \hline & & & & 0 & & \\ \hline & & & & 0 & 1 & 1 \\ & & & & 1 & 0 & 1 \\ & & & & 1 & 1 & 0 \\ \hline & & & & 0 & 0 & 0 & 0 \\ & & & & & & 1 & \\ \hline & & & & 0 & & & I \end{array} \right)$$

TAB. 4.7: Les points affinement indépendants pour $E_{\#}^1 = 3$, $p = 2$, $q_1 = 2$ et $\alpha'_{2,3} = \frac{1}{2}$

$$A_3 = \left(\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & & & \\ 1 & 0 & 1 & 1 & & & \\ 1 & 1 & 0 & 1 & & & \\ 1 & 1 & 1 & 0 & & & \\ \hline & & & & 0 & & \\ \hline & & & & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & & 0 & 1 & 1 & 0 & & & \\ & & & & 1 & 0 & 1 & 0 & & & \\ & & & & 1 & 1 & 0 & 1 & & & \\ \hline & & & & 1 & & & & 0 & 1 & 1 \\ & & & & & & & & 1 & 0 & 1 \\ & & & & & & & & 1 & 1 & 0 \end{array} \right)$$

TAB. 4.8: Les points affinement indépendants pour $E_{\#}^1 = 4$, $E_{\#}^2 = 3$, $p = 3$, $q_1 = 3$, $q_2 = 2$ et $\alpha_{4,3} = \frac{1}{5}$

4.1.3.1 Justification de l'algorithme

Dans le théorème suivant, nous allons établir les conditions nécessaires pour que les inégalités non dominées générées par l'algorithme 2 soient des facettes pour le polytope du sac à dos.

Théorème 4.1. *Toute inégalité non dominée générée par l'algorithme 2 définit une facette sur $\mathcal{P}_{C \cup E_1 \cup E_2}$ si les conditions suivantes sont satisfaites.*

- i) *L'inégalité domine une autre inégalité générée par l'algorithme 2;*
- ii) *Si $E_{\#}^1 - q_1 \geq 1$ alors*

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_1^1, j_{|E_1|-q_1+3}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\}$$

n'est pas une couverture;

- iii) *Si $E_{\#}^2 - q_2 \geq 2$ alors*

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_1^2, j_{|E_2|-q_2+2}^2, \dots, j_{|E_2|}^2\}$$

n'est pas une couverture;

- iv) *Si $E_{\#}^2 \neq E_2$, alors $\alpha_{E_{\#}^1, E_{\#}^2} > \alpha_{E_{\#}^1, E_{\#}^2+1}$.*

Preuve. Nous allons démontrer que les inégalités de la forme (4.10) générées par l'algorithme 2 et qui vérifient les conditions du théorème 4.1 sont des facettes.

Dans cette démonstration, nous allons utiliser la technique de la preuve directe de facettes, i.e. nous allons montrer que la dimension des faces définies par les inégalités non-dominées générées par l'algorithme 2 est $\dim(\mathcal{P}) - 1$

Les inégalités générées par l'algorithme 2 sont valides car ce dernier vérifie chaque point qui peut les rendre invalides.

Il est clair que la dimension de $\text{conv}(\mathcal{P}_{C \cup E_1 \cup E_2})$ est $|C| + |E_1| + |E_2|$ car l'origine ne satisfait pas toute inégalité de couverture simultanément liftée à égalité. Ainsi, d'après le théorème 2.1, la dimension de la face induite par toute inégalité :

$$\sum_{i \in C} x_i + \alpha_{E_{\#}^1, E_{\#}^2} \left(\sum_{j \in E_{\#}^1} x_j + \sum_{j \in E_{\#}^2} x_j \right) \leq |C| - 1 \quad (4.15)$$

générée par l'algorithme 2 peut être au plus $|C| + |E_1| + |E_2| - 1$.

Les $|C| + |E_1| + |E_2|$ points affinement indépendants sont déterminés comme suit :

- L'algorithme 2 exige que C soit minimal et ainsi le point $\sum_{i \in C \setminus \{h\}} e_i$ est faisable et satisfait l'inégalité 4.15 à égalité, pour chaque $h \in C$. D'où, nous avons $|C|$ points.
- Puisque l'inégalité (4.15) domine au moins une inégalité, donc $\alpha_{E_{\#}^1, E_{\#}^2 - 1} = \alpha_{E_{\#}^1, E_{\#}^2}$. Par conséquent, le point :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2}^{E_{\#}^2 - 1}} e_j$$

est faisable. Puisque l'ensemble E_2 est trié, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2+1}^{E_{\#}^2} \setminus \{k\}} e_j$$

sont faisables pour tout $k \in E_{q_2+1}^{E_{\#}^2}$. Nous obtenons $q_2 + 1$ points.

- De plus, si l'inégalité (4.15) domine une autre inégalité, l'ensemble :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2-x}^2, \dots, j_{|E_2|-x}^2\}$$

est une couverture minimale. D'où, l'ensemble :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2-x}^2, \dots, j_{|E_2|-x}^2\} \setminus \{l\}$$

où, $l \in L_1 = \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\}$ et $x \in \{1, \dots, q_2 - 1\}$, n'est pas une couverture minimale. Par conséquent, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1} \setminus \{l\}} e_j + \sum_{j \in E_{q_2+1}^{E_{\#}^2}} e_k$$

sont faisables. D'où, nous obtenons q_1 points.

- Par hypothèse, si $E_{\#}^1 - q_1 \geq 1$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_1^1, j_{|E_1|-q_1+3}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\}$$

n'est pas une couverture et puisque l'inégalité (4.15) domine une autre inégalité, alors les points :

$$\sum_{i \in C_p} e_i + e_l + \sum_{j \in E_{q_1-2}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2+1}^{E_{\#}^2}} e_j$$

où, $l = 1, \dots, E_{\#}^1 - q_1$, sont faisables. Donc, nous aurons $E_{\#}^1 - q_1$ points.

- Par hypothèse, nous avons, si $E_{\#}^2 - q_2 \geq 2$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_1^2, j_{|E_2|-q_2+2}^2, \dots, j_{|E_2|}^2\}$$

n'est pas une couverture. D'où, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + e_l + \sum_{j \in E_{q_2-1}^{E_{\#}^2}} e_j$$

où, $l = 1, \dots, E_{\#}^2 - q_2 - 1$, sont faisables. Ainsi, nous obtenons $E_{\#}^2 - q_2 - 1$ points.

- Si $E_{\#}^{E_2} \neq E_2$ et $\alpha_{E_{\#}^1, E_{\#}^2} > \alpha_{E_{\#}^1, E_{\#}^2+1}$, alors l'algorithme 2 change la valeur de $\alpha_{E_{\#}^1, E_{\#}^2}$ pendant la $(p, E_{\#}^1, E_{\#}^2 + 1)^{\text{ème}}$ itération itération. Par conséquent, le point :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2+1}^{E_{\#}^2+1}} e_j$$

est faisable. Ainsi, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2}^{E_{\#}^2}} e_j + e_l$$

sont faisables pour tout $l \in E_2 \setminus E_{\#}^{E_2}$. D'où, nous aurons $E_2 \setminus E_{\#}^{E_2}$ points.

Il est clair que ces points sont affinement indépendants, car la matrice de ces points comporte des sous matrices cycliquement permutées avec un seul zéro sur ces permutations. De plus, la matrice comporte aussi des lignes qui ont un seul un. \square

4.1.3.2 Complexité

La complexité de l'algorithme 2 est donnée par le théorème suivant :

Théorème 4.2. *La complexité de l'algorithme 2 est de l'ordre de :*

$$O(|C||E_1||E_2| + |C|\log|C| + |E_1|\log|E_1| + |E_2|\log|E_2|)$$

Preuve. Nous avons trois ensembles principaux, C (une couverture minimale), E_1 et E_2 ($E_1 \cup E_2 = N \setminus C$), ces ensembles sont triés dans un ordre décroissant de leurs coefficients w_j . La complexité de cette procédure (le trie) est de l'ordre de $O(|C|\log|C|)$, $O(|E_1|\log|E_1|)$ et $O(|E_2|\log|E_2|)$, respectivement.

L'algorithme prend un nombre constant p d'indices de la couverture minimale C et itère pour tous les indices : $1, \dots, |E_1|$. Dans chaque iteration et pour chacun de ces indices, l'algorithme itère pour tous les indices : $1, \dots, |E_2|$, d'où, ces deux boucles imbriquées sont de l'ordre de $O(|E_1||E_2|)$. Ces itérations sont alors répétées pour les valeurs variables de p , qui peuvent avoir un nombre maximum de $|C|$. Nous pouvons clairement voir que la complexité des trois boucles imbriquées est de l'ordre de $O(|C||E_1||E_2|)$.

Ainsi, l'algorithme 2 est de l'ordre de $O(|C||E_1||E_2| + |C|\log|C| + |E_1|\log|E_1| + |E_2|\log|E_2|)$. \square

4.2 Comparaison

Nous allons effectuer une comparaison entre l'algorithme 2 et MSLA, celle-ci est basée sur le déroulement d'un exemple.

Exemple 4.2. *Soit la contrainte du sac à dos suivante :*

$$29x_1 + 29x_2 + 27x_3 + 25x_4 + 14x_5 + 14x_6 + 12x_7 + 11x_8 + 11x_9 + 9x_{10} + 9x_{11} \leq 90. \quad (4.16)$$

La couverture de cette instance est $C = \{1, 2, 3, 4\}$ parce que la somme des coefficients $29 + 29 + 27 + 25 = 110$ est plus grande que le second membre de l'inégalité (4.16) : (39). C'est également une couverture minimale, alors l'inégalité minimale de couverture est $x_1 + x_2 + x_3 + x_4 \leq 3$. D'abord, nous allons partitionner l'ensemble $N \setminus C$ en :

- Pour MSLA : $E = \{5, 6, 7, 8, 9, 10, 11\}$;

- Pour l'algorithme 2 : $E_1 = \{5, 6, 7, 8\}$ et $E_2 = \{9, 10, 11\}$.

Les résultats du déroulement des algorithmes MSLA et algorithme 2 sont résumés dans les tableaux 4.9, 4.10 et 4.11 :

$E_{\#}$	$p=3$		$p=2$		$p=1$		$p=0$		$\min \alpha_{E_{\#}}^*$
	q	$\alpha'_{3,E_{\#}}$	q	$\alpha'_{2,E_{\#}}$	q	$\alpha'_{1,E_{\#}}$	q	$\alpha'_{0,E_{\#}}$	
1	1	∞	1	1	1	2	1	3	1
2	1	∞	2	$\frac{1}{2}$	2	1	2	$\frac{3}{2}$	$\frac{1}{2}$
3	1	∞	3	$\frac{1}{2}$	3	$\frac{2}{3}$	3	1	$\frac{1}{2}$
4	1	∞	3	$\frac{1}{3}$	4	$\frac{1}{2}$	4	$\frac{3}{4}$	$\frac{1}{3}$
5	1	∞	4	$\frac{1}{3}$	5	$\frac{2}{5}$	5	$\frac{3}{5}$	$\frac{1}{3}$
6	1	0	4	$\frac{1}{3}$	6	$\frac{2}{5}$	6	$\frac{1}{2}$	0
7	1	0	4	$\frac{1}{3}$	6	$\frac{2}{5}$	7	$\frac{3}{7}$	0

TAB. 4.9: Résultats générés par MSLA

$E_{\#}^1$	$p=3$		$p=2$		$p=1$		$p=0$		$\min \alpha_{p,E_{\#}^1}^*$
	q	$\alpha'_{3,E_{\#}^1}$	q	$\alpha'_{2,E_{\#}^1}$	q	$\alpha'_{1,E_{\#}^1}$	q	$\alpha'_{0,E_{\#}^1}$	
1	1	∞	1	1	1	2	1	3	1
2	1	∞	2	$\frac{1}{2}$	2	1	2	$\frac{3}{2}$	$\frac{1}{2}$
3	1	∞	3	$\frac{1}{2}$	3	$\frac{2}{3}$	3	1	$\frac{1}{2}$
4	1	∞	3	$\frac{1}{3}$	4	$\frac{1}{2}$	4	$\frac{3}{4}$	$\frac{1}{3}$

TAB. 4.10: Les valeurs de $\alpha'_{p,E_{\#}^1}$ générées par l'algorithme 2

$E_{\#}^1 \backslash E_{\#}^2$	$p=2$				$p=1$				$p=0$				$\min \alpha_{E_{\#}^1, E_{\#}^2}^*$
	1	2	3	4	1	2	3	4	1	2	3	4	
1	$\frac{1}{2}$	$\frac{1}{2}$	-	$\frac{1}{3}$	1	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{3}{2}$	1	$\frac{3}{4}$	$\frac{3}{5}$	$\frac{1}{3}$
2	$\frac{1}{3}$	$\frac{1}{3}$	-	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{2}{5}$	1	$\frac{3}{4}$	$\frac{3}{5}$	$\frac{1}{2}$	$\frac{1}{3}$
3	$\frac{1}{3}$	$\frac{1}{3}$	-	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{3}{4}$	$\frac{3}{5}$	$\frac{1}{2}$	$\frac{3}{7}$	$\frac{1}{3}$

TAB. 4.11: Les valeurs de $\alpha_{E_{\#}^1, E_{\#}^2}$ générées par l'algorithme 2

Remarque 4.1. Dans le tableau 4.12 suivant nous avons toutes les inégalités non-dominées générées par MSLA et l'algorithme 2 qui représentent des facettes.

MSLA	l'algorithme 2
$\sum_{i=1}^4 x_i + 1x_5 \leq 3$	$\sum_{i=1}^4 x_i + 1x_5 \leq 3$
$\sum_{i=1}^4 x_i + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$	$\sum_{i=1}^4 x_i + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$
$\sum_{i=1}^4 x_i + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$	$\sum_{i=1}^4 x_i + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$

TAB. 4.12: Les inégalités générées par MSLA et l'algorithme 2

Nous remarquons que l'algorithme 2 génère l'inégalité :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 3$$

qui domine l'inégalité générée par MSLA :

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7 + x_8 + x_9) \leq 3$$

D'après cet exemple, nous remarquons que contrairement à MSLA, l'algorithme 2 permet de générer des inégalités non-dominées maximales même si les coefficients w_j des indices appartenants à l'ensemble $N \setminus C$ sont inférieurs, ou égaux, à $d - \sum_{j=2}^{|C|} w_j$, à condition que, ces coefficients n'appartiennent pas à l'ensemble E_1 (il peuvent appartenir à l'ensemble E_2).

Conclusion

Dans ce chapitre, nous avons développé un algorithme (algorithme 2) qui permet de lifter simultanément deux ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Les inégalités générées par cet algorithme sont valides et d'une grande dimension. Nous avons également énoncé les conditions nécessaires pour que les inégalités non-dominées qu'il génère soient des facettes pour \mathcal{P} . Par la suite, nous avons effectué une comparaison, basée sur le déroulement d'un exemple, entre cet algorithme et MSLA.

Est ce qu'il est possible de généraliser le lifting simultané sur plusieurs ensembles de variables ?

Lifting simultané de plusieurs ensembles de variables

L’objectif de ce chapitre est de généraliser le lifting simultané sur plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. D’abord, nous commençons par développer un algorithme qui permet de lifter simultanément trois ensembles de variables en une inégalité de couverture, puis, nous énonçons les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du problème de sac à dos. Nous terminons par une proposition de la généralisation du lifting simultané sur plusieurs ensembles de variables en développant un algorithme qui permet de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture en y introduisant les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du problème de sac à dos.

5.1 Lifting simultané de trois ensembles de variables

Pour le lifting simultané de trois ensembles de variables E_1 , E_2 et E_3 , nous allons opérer de la même manière que pour le lifting simultané de deux ensembles de variables présenté dans le chapitre précédent, puis, développer un algorithme qui permet de lifter simultanément trois ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Les inégalités générées par cet algorithme peuvent avoir les formes (4.1), (4.10) et la forme suivante :

$$\sum_{i \in C} x_i + \alpha^1 \sum_{j \in E_1} x_j + \alpha^2 \sum_{j \in E_2} x_j + \alpha^3 \sum_{j \in E_3} x_j \leq |C| - 1 \quad (5.1)$$

Sachant que les quatre ensembles C , E_1 , E_2 et E_3 sont triés selon un ordre décroissant de leurs coefficients w_j tels que : $w_i \leq w_{i+1}$, $w_{j^1} \leq w_{j^1+1}$, $w_{j^2} \leq w_{j^2+1}$ et $w_{j^3} \leq w_{j^3+1}$ pour C , E_1 , E_2 et E_3 respectivement.

5.1.1 Principe de l'algorithme

Nous suivons le même principe que l'algorithme 2 i.e. parcourir l'ensemble E_1 et prendre un nombre de variables, noté q_1 , pour les lifter. Puis et à chaque itération de ce parcours voir si c'est possible ($T_1 \leq d$) de parcourir l'ensemble E_2 et de prendre un nombre de variables, noté q_2 , et les lifter à leurs tour. Dans l'algorithme 3 nous opérons de la même manière et à chaque itération du parcours de l'ensemble E_2 nous testons si c'est possible ($T_2 \leq d$) de parcourir l'ensemble E_3 et de prendre un nombre de variables, noté q_3 , pour les lifter. Pour éviter la répétition, nous décrivons uniquement l'étape où nous parcourons le troisième ensemble E_3 que nous rajoutons à l'étape 1.1 présentée dans la section 4.1.1.

Nous choisissons le premier élément j_1^3 de l'ensemble E_3 , et nous mettons la valeur de E_Σ^3 à $w_{j_1^3}$, $E_\#^3 = 1$, $q_3 = 1$ et $\alpha^1 = \alpha^2 = \alpha^3 = \alpha'$. Nous calculons le troisième total $T_3 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2 + E_\Sigma^3$:

Étape 1.1.1 : Si $T_3 \leq d$, alors p éléments de C , q_1 éléments de $\{j_1^1, \dots, j_{E_\#^1}^1\}$, q_2 éléments de $\{j_1^2, \dots, j_{E_\#^2}^2\}$ et q_3 éléments de $\{j_1^3, \dots, j_{E_\#^3}^3\}$ ne forment pas une couverture, donc il existe le point faisable correspondant et les valeurs des coefficients α^1 , α^2 et α^3 doivent être changées de sorte que ce point avec p , q_1 , q_2 et q_3 éléments satisfait l'inégalité (5.1) à égalité. Les valeurs de q_3 et $E_\#^3$ sont incrémentées de 1.

Étape 1.1.2 : Si $T_3 > d$, alors les p éléments de C , q_1 éléments de $\{j_1^1, \dots, j_{E_\#^1}^1\}$, q_2 éléments de $\{j_1^2, \dots, j_{E_\#^2}^2\}$ et q_3 éléments de $\{j_1^3, \dots, j_{E_\#^3}^3\}$ forment une couverture et donc les coefficients du lifting α^1 , α^2 et α^3 restent inchangés ($\alpha_{E_\#, E_\#}^1 = \alpha_{E_\#, E_\#-1}^1$, $\alpha_{E_\#, E_\#}^2 =$

$\alpha_{E_{\#}^2, E_{\#}^3-1}^2$ et $\alpha_{E_{\#}^2, E_{\#}^3}^3 = \alpha_{E_{\#}^2, E_{\#}^3-1}^3$). Par conséquent, uniquement $E_{\#}^3$ sera incrémenter de 1.

Cette alternative sera répétée jusqu'à ce que $E_{\#}^3$ soit égal à $|E_3|$.

5.1.2 Calcul des coefficients α^1 , α^2 et α^3

Le calcul des coefficients α^1 , α^2 et α^3 pour les inégalités de la forme (5.1), se fait de la manière suivante :

Si $T_3 \leq d$, cela implique que : $C_{\Sigma} + E_{\Sigma}^1 \leq d$ (resp : $C_{\Sigma} + E_{\Sigma}^2 \leq d$, $C_{\Sigma} + E_{\Sigma}^3 \leq d$), alors p éléments de C , q_1 (resp : q_2, q_3) éléments de $\{j_1^1, \dots, j_{E_{\#}^1}^1\}$ (resp : $\{j_1^2, \dots, j_{E_{\#}^2}^2\}, \{j_1^3, \dots, j_{E_{\#}^3}^3\}$) ne forment pas une couverture, donc il existe le point faisable correspondant. Cependant la valeur du coefficient α^1 (resp : α^2, α^3) doit être changée de sorte que ce point avec p et q_1 (resp : q_2, q_3) éléments satisfait l'inégalité (5.1) à égalité. Donc, nous avons :

$$\alpha^1 = \frac{|C| - 1 - p}{q_1} \quad (5.2)$$

$$\alpha^2 = \frac{|C| - 1 - p}{q_2} \quad (5.3)$$

$$\alpha^3 = \frac{|C| - 1 - p}{q_3} \quad (5.4)$$

Par substitution dans l'inégalité (5.1), pour $\sum_{i \in C} x_i = p$, $\sum_{j \in E_1} x_j = q_1$, $\sum_{j \in E_2} x_j = q_2$ et $\sum_{j \in E_3} x_j = q_3$, nous aurons :

$$\frac{|C| - 1 - p}{q_1} \times q_1 + \frac{|C| - 1 - p}{q_2} \times q_2 + \frac{|C| - 1 - p}{q_3} \times q_3 = 3(|C| - 1 - p) > |C| - 1 - p |C| - 1$$

En multipliant les coefficients α^1 , α^2 et α^3 par λ_1 , λ_2 et λ_3 respectivement, nous aurons :

$$\alpha^1 = \frac{|C| - 1 - p}{q_1} \times \lambda_1 \quad (5.5)$$

$$\alpha^2 = \frac{|C| - 1 - p}{q_2} \times \lambda_2 \quad (5.6)$$

$$\alpha^3 = \frac{|C| - 1 - p}{q_3} \times \lambda_3 \quad (5.7)$$

où, $\lambda_1, \lambda_2, \lambda_3 \in]0, 1[$ et par substitution dans l'inégalité (5.1) :

$$\frac{|C| - 1 - p}{q_1} \times \lambda_1 \times q_1 + \frac{|C| - 1 - p}{q_2} \times \lambda_2 \times q_2 + \frac{|C| - 1 - p}{q_3} \times \lambda_3 \times q_3 = (\lambda_1 + \lambda_2 + \lambda_3)(|C| - 1 - p)$$

Pour que l'inégalité (5.1) soit satisfaite à égalité, nous devons avoir :

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \quad (5.8)$$

Nous avons choisi de prendre :

- $\lambda_1 = \frac{q_1}{q_1 + q_2 + q_3}$: Le taux des éléments de E_1 pris dans $N \setminus C$.
- $\lambda_2 = \frac{q_2}{q_1 + q_2 + q_3}$: Le taux des éléments de E_2 pris dans $N \setminus C$.
- $\lambda_3 = \frac{q_3}{q_1 + q_2 + q_3}$: Le taux des éléments de E_3 pris dans $N \setminus C$.

D'où :

$$\left. \begin{aligned} \alpha^1 &= \frac{|C| - 1 - p}{q_1} \times \frac{q_1}{q_1 + q_2 + q_3} = \frac{|C| - 1 - p}{q_1 + q_2 + q_3} \\ \alpha^2 &= \frac{|C| - 1 - p}{q_2} \times \frac{q_2}{q_1 + q_2 + q_3} = \frac{|C| - 1 - p}{q_1 + q_2 + q_3} \\ \alpha^3 &= \frac{|C| - 1 - p}{q_3} \times \frac{q_3}{q_1 + q_2 + q_3} = \frac{|C| - 1 - p}{q_1 + q_2 + q_3} \end{aligned} \right\} \Rightarrow \alpha^1 = \alpha^2 = \alpha^3 = \frac{|C| - 1 - p}{q_1 + q_2 + q_3} \quad (5.9)$$

5.1.3 Formes des inégalités générées

Puisque $\alpha^1 = \alpha^2 = \alpha^3$, alors les inégalités générées par l'algorithme 3 sont de la forme (4.1), (4.10) ou bien la forme suivante :

$$\sum_{i \in C} x_i + \beta \left(\sum_{j \in E_1} x_j + \sum_{j \in E_2} x_j + \sum_{j \in E_3} x_j \right) \leq |C| - 1 \quad (5.10)$$

où, $\beta = \alpha^1 = \alpha^2 = \alpha^3$

Algorithme 3 : Algorithme pour le lifting simultané de trois ensembles de variables

Données : Une contrainte du problème du sac à dos : $\sum_{j \in N} w_j x_j \leq d$.

Les ensembles de variables C, E_1, E_2 et E_3 ($N \setminus C = E_1 \cup E_2 \cup E_3$).

Résultats : Des inégalités de la forme (4.1), (4.10) et (5.10)

$p := |C| - 1$;

while $p \geq 0$ **do**

$C_\Sigma := \sum_{r=|C|-p+1}^{|C|} w_{i_r}$; $\alpha'_{p,0} := \infty$; $q_1 := 1$; $E_\#^1 := 1$;

while $E_\#^1 \leq |E_1|$ **do**

$E_\Sigma^1 := \sum_{r=E_\#^1-q_1+1}^{E_\#^1} w_{j_r^1}$; $T_1 = C_\Sigma + E_\Sigma^1$;

if $T_1 \leq c$ **then**

$\alpha'_{p,E_\#^1} := \frac{|C|-1-p}{q_1}$; $q_1 := q_1 + 1$; $\alpha_{E_\#^1,0} := \alpha'_{p,E_\#^1}$ $q_2 := 1$; $E_\#^2 := 1$;

while $E_\#^2 \leq |E_2|$ **do**

$E_\Sigma^2 := \sum_{r=E_\#^2-q_2+1}^{E_\#^2} w_{j_r^2}$; $T_2 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2$;

if $T_2 \leq c$ **then**

$\alpha_{E_\#^1,E_\#^2} := \frac{|C|-1-p}{q_1+q_2}$; $q_2 := q_2 + 1$; $\beta_{E_\#^2,0} := \alpha_{E_\#^1,E_\#^2}$;

$q_3 := 1$; $E_\#^3 := 1$;

while $E_\#^3 \leq |E_3|$ **do**

$E_\Sigma^3 := \sum_{r=E_\#^3-q_3+1}^{E_\#^3} w_{j_r^3}$; $T_3 = C_\Sigma + E_\Sigma^1 + E_\Sigma^2 + E_\Sigma^3$;

if $T_3 \leq c$ **then**

$\beta_{E_\#^2,E_\#^3} := \frac{|C|-1-p}{q_1+q_2+q_3}$; $q_3 := q_3 + 1$;

else

$\beta_{E_\#^2,E_\#^3} := \beta_{E_\#^2,E_\#^3-1}$;

$E_\#^3 := E_\#^3 + 1$;

else

$\alpha_{E_\#^1,E_\#^2} := \alpha_{E_\#^1,E_\#^2-1}$;

$E_\#^2 := E_\#^2 + 1$;

else

$\alpha'_{p,E_\#^1} := \alpha'_{p,E_\#^1-1}$;

$E_\#^1 := E_\#^1 + 1$;

$p := p - 1$;

Exemple 5.1. Soit la contrainte du sac à dos suivante :

$$37x_1 + 33x_2 + 20x_3 + 15x_4 + 11x_5 + 8x_6 + 4x_7 + 2x_8 + 2x_9 + 1x_{10} + 1x_{11} + 1x_{12} \leq 39. \quad (5.11)$$

$E_1^\#$	inégalité E_1	$E_2^\#$	$E_3^\#$	q_3	$E_3^\#$	T_3	β	inégalité $E_1 \cup E_2$	inégalité $E_1 \cup E_2 \cup E_3$
$q_1 = 1$ w_3 $E_2^1 = 20$ $T_1 = 20$	$1 \leq \sum_{i=1}^2 x_i + 1x_3$	$q_2 = 1$ w_7	$\alpha_{1,1} = \frac{1}{2}$	1	1	25	$\frac{3}{5}$	$x_1 + x_2 + \frac{1}{2}(x_3 + x_7) \leq 1$	$x_1 + x_2 + (1/3)(x_3 + x_7 + x_{10}) \leq 1$
			$E_2^1 = 4$	2	2	26	$\frac{4}{4}$		$x_1 + x_2 + (1/4)(x_3 + x_7 + \sum_{i=10}^4 x_i) \leq 1$
			$T_2 = 24$	3	3	27	$\frac{1}{5}$		$x_1 + x_2 + (1/5)(x_3 + x_7 + \sum_{i=10}^4 x_i) \leq 1$
$\alpha_{0,1} = 1$	$1 \leq \sum_{i=1}^2 x_i + 1x_3$	$q_2 = 2$ w_7, w_8	$\alpha_{1,2} = \frac{3}{5}$	1	1	27	$\frac{4}{4}$	$x_1 + x_2 + \frac{1}{3}(x_3 + \sum_{i=7}^8 x_i) \leq 1$	$x_1 + x_2 + (1/4)(x_3 + \sum_{i=7}^8 x_i + x_{10}) \leq 1$
			$E_2^1 = 6$	2	2	28	$\frac{5}{5}$		$x_1 + x_2 + (1/5)(x_3 + \sum_{i=7}^8 x_i + \sum_{i=10}^{11} x_i) \leq 1$
			$T_2 = 26$	3	3	29	$\frac{1}{6}$		$x_1 + x_2 + (1/6)(x_3 + \sum_{i=7}^8 x_i + \sum_{i=10}^{12} x_i) \leq 1$
$q_1 = 2$ w_3, w_4 $E_2^1 = 20$ $T_1 = 35$	$1 \leq \sum_{i=1}^2 x_i + \sum_{i=3}^5 x_i$	$q_2 = 1$ w_7	$\alpha_{1,1} = \frac{1}{3}$	1	1	40	$\frac{4}{4}$	$x_1 + x_2 + \frac{1}{5}(x_3 + \sum_{i=7}^9 x_i) \leq 1$	$x_1 + x_2 + (1/4)(\sum_{i=3}^4 x_i + x_7 + x_{10}) \leq 1$
			$E_2^1 = 4$	2	1	40	$\frac{1}{4}$		$x_1 + x_2 + (1/4)(\sum_{i=3}^4 x_i + x_7 + x_{10} + \sum_{i=11} x_i) \leq 1$
			$T_2 = 39$	3	1	40	$\frac{1}{4}$		$x_1 + x_2 + (1/4)(\sum_{i=3}^4 x_i + x_7 + x_{10} + \sum_{i=10}^{12} x_i) \leq 1$
$\alpha_{0,2} = \frac{1}{2}$	$1 \leq \sum_{i=1}^2 x_i + \sum_{i=3}^4 x_i$	$q_2 = 2$ w_7, w_8	$\alpha_{1,2} = \frac{1}{3}$	1	1	40	$\frac{4}{4}$	$x_1 + x_2 + \frac{1}{5}(\sum_{i=3}^4 x_i + \sum_{i=7}^8 x_i) \leq 1$	$x_1 + x_2 + (1/4)(\sum_{i=3}^4 x_i + \sum_{i=7}^9 x_i + x_{10}) \leq 1$
			$E_2^1 = 6$	2	1	40	$\frac{1}{4}$		$x_1 + x_2 + (1/4)(\sum_{i=3}^4 x_i + \sum_{i=7}^9 x_i + \sum_{i=10}^{11} x_i) \leq 1$
			$T_2 = 41$	3	1	40	$\frac{1}{4}$		$x_1 + x_2 + (1/4)(\sum_{i=3}^4 x_i + \sum_{i=7}^9 x_i + \sum_{i=10}^{12} x_i) \leq 1$
$q_1 = 3$ w_3, w_4 w_5 $E_2^1 = 46$ $T_1 = 46$ $\alpha_{0,3} = \frac{1}{2}$	$1 \leq \sum_{i=1}^2 x_i + \sum_{i=3}^5 x_i$	$q_2 = 1$ w_7	$\alpha_{1,1} = \frac{1}{4}$	1	1	39	$\frac{1}{5}$	$x_1 + x_2 + \frac{1}{4}(\sum_{i=3}^6 x_i + x_7) \leq 1$	$x_1 + x_2 + (1/5)(\sum_{i=3}^6 x_i + x_7 + x_{10}) \leq 1$
			$E_2^1 = 4$	2	2	40	$\frac{1}{5}$		$x_1 + x_2 + (1/5)(\sum_{i=3}^6 x_i + x_7 + \sum_{i=10}^{11} x_i) \leq 1$
			$T_2 = 38$	3	2	40	$\frac{1}{5}$		$x_1 + x_2 + (1/5)(\sum_{i=3}^6 x_i + x_7 + \sum_{i=10}^{12} x_i) \leq 1$
$q_1 = 3$ w_4, w_5 w_6 $E_2^1 = 34$ $T_1 = 34$ $\alpha_{0,4} = \frac{1}{3}$	$1 \leq \sum_{i=1}^2 x_i + \sum_{i=3}^6 x_i$	$q_2 = 2$ w_7, w_8	$\alpha_{1,2} = \frac{1}{4}$					$x_1 + x_2 + \frac{1}{4}(\sum_{i=3}^6 x_i + \sum_{i=7}^8 x_i) \leq 1$	$x_1 + x_2 + (1/6)(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i + x_{10}) \leq 1$
			$E_2^1 = 6$						$x_1 + x_2 + (1/6)(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i + \sum_{i=10}^{11} x_i) \leq 1$
			$T_2 = 40$						$x_1 + x_2 + (1/6)(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i + \sum_{i=10}^{12} x_i) \leq 1$
$q_1 = 3$ w_4, w_5 w_6 $E_2^1 = 34$ $T_1 = 34$ $\alpha_{0,4} = \frac{1}{3}$	$1 \leq \sum_{i=1}^2 x_i + \sum_{i=3}^6 x_i$	$q_2 = 2$ w_8, w_9	$\alpha_{1,3} = \frac{1}{5}$	1	1	39	$\frac{1}{6}$	$x_1 + x_2 + \frac{1}{5}(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i) \leq 1$	$x_1 + x_2 + (1/6)(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i + x_{10}) \leq 1$
			$E_2^1 = 4$	2	2	40	$\frac{1}{6}$		$x_1 + x_2 + (1/6)(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i + \sum_{i=10}^{11} x_i) \leq 1$
			$T_2 = 38$	3	2	40	$\frac{1}{6}$		$x_1 + x_2 + (1/6)(\sum_{i=3}^6 x_i + \sum_{i=7}^9 x_i + \sum_{i=10}^{12} x_i) \leq 1$

TAB. 5.1: Les résultats du déroulement de l'exemple 5.1, pour $p = 0$ et $G_2 = 0$, avec $C = \{1, 2\}$, $E_1 = \{3, 4, 5, 6\}$, $E_2 = \{7, 8, 9\}$ et $E_3 = \{10, 11, 12\}$

Les valeurs de $\alpha'_{p,E_{\#}}, \alpha_{E_{\#},E_{\#}^2}$ et $\beta_{E_{\#},E_{\#}^3}$ générées par l'algorithme 3 sont résumés dans les tableaux 5.2, 5.3 et 5.4 respectivement.

	$p = 1$		$p = 0$		
$E_{\#}^1$	q_1	$\alpha'_{3,E_{\#}^1}$	q_1	$\alpha'_{2,E_{\#}^1}$	$\min \alpha_{p,E_{\#}^1}^*$
1	1	∞	1	1	1
2	1	∞	2	$\frac{1}{2}$	$\frac{1}{2}$
3	1	∞	3	$\frac{1}{2}$	$\frac{1}{2}$
4	1	∞	3	$\frac{1}{2}$	$\frac{1}{3}$

TAB. 5.2: Les valeurs de $\alpha'_{p,E_{\#}^1}$ générées par l'algorithme 3

		$p = 0$				
$E_{\#}^2$	$E_{\#}^1$	1	2	3	4	$\min \alpha_{E_{\#}^1,E_{\#}^2}^*$
1	1	$\frac{1}{2}$	$\frac{1}{3}$	-	$\frac{1}{4}$	$\frac{1}{4}$
2	1	$\frac{1}{3}$	$\frac{1}{3}$	-	$\frac{1}{4}$	$\frac{1}{4}$
3	1	$\frac{1}{4}$	$\frac{1}{4}$	-	$\frac{1}{5}$	$\frac{1}{5}$

TAB. 5.3: Les valeurs de $\alpha_{E_{\#},E_{\#}^2}$ générées par l'algorithme 3

		$p = 0$												
		$E_{\#}^1 = 1$			$E_{\#}^1 = 2$			$E_{\#}^1 = 3$			$E_{\#}^1 = 4$			$\min \beta_{E_{\#}^2,E_{\#}^3}^*$
$E_{\#}^3$	$E_{\#}^2$	1	2	3	1	2	3	1	2	3	1	2	3	
1	1	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{4}$	-	$\frac{1}{4}$	-	-	-	$\frac{1}{5}$	-	$\frac{1}{6}$	$\frac{1}{6}$
2	1	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{4}$	-	$\frac{1}{4}$	-	-	-	$\frac{1}{5}$	-	$\frac{1}{6}$	$\frac{1}{6}$
3	1	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{4}$	-	$\frac{1}{4}$	-	-	-	$\frac{1}{5}$	-	$\frac{1}{6}$	$\frac{1}{7}$

TAB. 5.4: Les valeurs de $\beta_{E_{\#},E_{\#}^3}$ générées par l'algorithme 3

L'algorithme 3 génère des inégalités dont plusieurs sont dominées par d'autres inégalités. Par exemple, l'inégalité $x_1 + x_2 + \frac{1}{4}(x_3 + x_4 + x_5 + x_6 + x_7) \leq 1$ est dominé par l'inégalité $x_1 + x_2 + \frac{1}{4}(x_3 + x_4 + x_5 + x_6 + x_7 + x_8) \leq 1$. Ainsi, les inégalités non-dominées sont les inégalités avec le plus grand nombre de variables pour une valeur particulière de α, α' et β . Dans cet exemple, il y a quatre inégalités non-dominées :

$$x_1 + x_2 + 1(x_3) \leq 1 \tag{5.12}$$

$$x_1 + x_2 + \frac{1}{2}(x_3 + x_4 + x_5) \leq 1 \tag{5.13}$$

$$x_1 + x_2 + \frac{1}{4}(x_3 + x_4 + x_5 + x_6 + x_7 + x_8) \leq 1 \quad (5.14)$$

$$x_1 + x_2 + \frac{1}{6}(x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11}) \leq 1 \quad (5.15)$$

- L'inégalité (5.12) correspond au minimum des $\alpha'_{p,E_{\#}^1} = 1$ dans le tableau 5.2, pour $E_{\#}^1 = 1$, $p = 0$ et $q_1 = 1$.
- L'inégalité (5.13) correspond au minimum des $\alpha'_{p,E_{\#}^1} = \frac{1}{2}$ dans le tableau 5.2, pour $E_{\#}^1 = 3$, $p = 0$ et $q_1 = 2$.
- L'inégalité (5.14) correspond au minimum des $\alpha_{E_{\#}^1, E_{\#}^2} = \frac{1}{4}$ dans le tableau 5.3, pour $E_{\#}^1 = 4$, $E_{\#}^2 = 2$, $p = 0$, $q_1 = 3$ et $q_2 = 1$.
- L'inégalité (5.15) correspond au minimum des $\beta_{E_{\#}^2, E_{\#}^3} = \frac{1}{6}$ dans le tableau 5.4, pour $E_{\#}^1 = 4$, $E_{\#}^2 = 3$, $E_{\#}^3 = 2$, $p = 0$, $q_1 = 3$, $q_2 = 2$ et $q_3 = 1$.

De l'exemple ci-dessus, il est clair que l'algorithme 3 génère des inégalités liftées maximales fortes. Les inégalités (5.12), (5.13), (5.14) et (5.15) représentent des facettes, car nous pouvons exhiber 11 point affinement indépendants pour chacune d'elles, ces derniers peuvent être représentés par les matrices A_4 , A_5 , A_6 et A_7 respectivement.

$$A_4 = \left(\begin{array}{c|c|c} \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} & & \mathbf{0} \\ \hline & \begin{array}{cccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} & \\ \hline \mathbf{0} & & I \end{array} \right)$$

$$A_5 = \left(\begin{array}{c|c|c} \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} & & \mathbf{0} \\ \hline & \begin{array}{ccc|cccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} & \\ \hline & \begin{array}{ccc|cccc} 1 & 0 & 1 & & & & & & & & \end{array} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & & & & & & & & & I \end{array} \right)$$

Les points affinement indépendants pour $E_{\#}^1 = 1$, $q_1 = 1$, $p = 0$ et $\alpha'_{0,1} = 1$

Les points affinement indépendants pour $E_{\#}^1 = 3$, $q_1 = 2$, $p = 0$ et $\alpha'_{0,3} = \frac{1}{2}$

$$A_6 = \left(\begin{array}{c|c|c} \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} & & \mathbf{0} \\ \hline & \begin{array}{ccc|c|cccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} & \\ \hline & \begin{array}{ccc|c|cc} 0 & 1 & 1 & 0 & \mathbf{1} & & \mathbf{1} & & & & \end{array} & \\ \hline & \begin{array}{ccc|c|cc} 1 & 0 & 1 & 0 & & & & & & & \end{array} & \\ \hline & \begin{array}{ccc|c|cc} 1 & 1 & 0 & 1 & & & & & & & \end{array} & \\ \hline \mathbf{0} & \mathbf{1} & & 1 & 1 & 0 & & & & & 1 \\ \hline & & & 1 & 0 & 1 & & & & & 0 \\ \hline \mathbf{0} & & & 0 & & & & & & & I \\ \hline & & & 0 & & & & & & & \\ \hline & & & 0 & & & & & & & \\ \hline & & & 0 & & & & & & & \\ \hline & & & 0 & & & & & & & \end{array} \right)$$

$$A_7 = \left(\begin{array}{c|c|c} \begin{array}{cc} 0 & 1 \\ 1 & 1 \end{array} & & \mathbf{0} \\ \hline & \begin{array}{ccc|c|cccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} & \\ \hline & \begin{array}{ccc|c|cc} 0 & 1 & 1 & 0 & \mathbf{1} & & \mathbf{1} & & & 1 \end{array} & \\ \hline & \begin{array}{ccc|c|cc} 1 & 0 & 1 & 0 & & & & & & 1 \end{array} & \\ \hline & \begin{array}{ccc|c|cc} 1 & 1 & 0 & 1 & & & & & & 1 \end{array} & \\ \hline & \mathbf{0} & & 0 & & & & & & & 0 \\ \hline & \mathbf{1} & & 1 & & & I & & & & \mathbf{1} \\ \hline & \mathbf{1} & & 1 & & & \mathbf{1} & & 1 & 0 & 0 \\ \hline & & & 1 & & & & & 0 & 1 & 1 \\ \hline & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right)$$

Les points affinement indépendants pour $E_{\#}^1 = 4$, $E_{\#}^2 = 2$, $q_1 = 3$, $p = 0$, $q_2 = 1$ et $\alpha_{4,2} = \frac{1}{4}$

Les points affinement indépendants pour $E_{\#}^1 = 4$, $E_{\#}^2 = 3$, $E_{\#}^3 = 2$, $q_1 = 3$, $p = 0$, $q_2 = 2$, $q_3 = 1$ et $\beta_{3,2} = \frac{1}{6}$

5.1.4 Justification de l'algorithme

Dans le théorème suivant, nous allons établir les conditions nécessaires pour que les inégalités non dominées générées par l'algorithme 3 soient des facettes pour le polytope du sac à dos.

Théorème 5.1. *Toute inégalité non dominée générée par l'algorithme 3 définit une facette sur $\mathcal{P}_{C \cup E_1 \cup E_2 \cup E_3}$, si les conditions suivantes sont satisfaites :*

i) *L'inégalité domine une autre inégalité générée par l'algorithme 3;*

ii) *Si $E_{\#}^1 - q_1 \geq 1$, alors l'ensemble :*

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_1^1, j_{|E_1|-q_1+3}, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3+1}, \dots, j_{|E_3|}^3\}$$

n'est pas une couverture;

iii) *Si $E_{\#}^2 - q_2 \geq 1$, alors l'ensemble :*

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}, \dots, j_{|E_1|}^1\} \cup \{j_1^2, j_{|E_2|-q_2+3}, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3+1}, \dots, j_{|E_3|}^3\}$$

n'est pas une couverture;

iv) *Si $E_{\#}^3 - q_3 \geq 2$, alors l'ensemble :*

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}, \dots, j_{|E_2|}^2\} \cup \{j_1^3, j_{|E_3|-q_3+2}, \dots, j_{|E_3|}^3\}$$

n'est pas une couverture;

v) *Si $E_{\#}^3 \neq E_3$, alors $\beta_{E_{\#}^2, E_{\#}^3} > \beta_{E_{\#}^2, E_{\#}^3+1}$.*

Preuve. Nous allons démontrer que les inégalités de la forme (5.10) générées par l'algorithme 3 et qui vérifient les conditions du théorème 5.1 sont des facettes.

Dans cette démonstration, nous allons utiliser la technique de la preuve directe de facettes, i.e. nous allons montrer que la dimension des faces définies par les inégalités générées par l'algorithme 3 est $\dim(\mathcal{P}) - 1$.

Les inégalités générées par l'algorithme 3 sont valides car ce dernier vérifie chaque point qui peut les rendre invalides.

Il est clair que la dimension de $\text{conv}(\mathcal{P}_{C \cup E_1 \cup E_2 \cup E_3})$ est $|C| + |E_1| + |E_2| + |E_3|$, car l'origine ne satisfait pas toute inégalité de couverture simultanément liftée à égalité. Ainsi, d'après le théorème 2.1, la dimension de la face induite par toute inégalité :

$$\sum_{i \in C} x_i + \beta_{E_{\#}^2, E_{\#}^3} \left(\sum_{j \in E_{\#}^1} x_j + \sum_{j \in E_{\#}^2} x_j + \sum_{j \in E_{\#}^3} x_j \right) \leq |C| - 1 \quad (5.16)$$

générée par l'algorithme 3 peut être au plus égale à $|C| + |E_1| + |E_2| + |E_3| - 1$.

Les $|C| + |E_1| + |E_2| + |E_3|$ points affinement indépendants sont déterminés comme suit :

- L'algorithme 3 exige que C soit minimal, ainsi, le point $\sum_{i \in C \setminus \{h\}} e_i$ est faisable et satisfait l'inégalité (5.16) à égalité, pour chaque $h \in C$. D'où, nous avons $|C|$ points.
- Puisque l'inégalité (5.16) domine au moins une inégalité, donc $\beta_{E_{\#}^2, E_{\#}^3-1} = \beta_{E_{\#}^2, E_{\#}^3}$. Par conséquent, le point :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2}^{E_{\#}^2}} e_j + \sum_{j \in E_{q_3}^{E_{\#}^3-1}} e_j$$

est faisable. Puisque l'ensemble E_3 est trié, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2}^{E_{\#}^2}} e_j + \sum_{j \in E_{q_3+1}^{E_{\#}^3} \setminus \{h\}} e_j$$

sont faisables, pour tout $h \in E_{q_3+1}^{E_{\#}^3}$. Ainsi, nous obtenons $q_3 + 1$ points.

- De plus, si l'inégalité (5.16) domine une autre inégalité, l'ensemble :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3-x}^3, \dots, j_{|E_3|-x}^3\}$$

est une couverture minimale, alors les ensembles :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3-x}^3, \dots, j_{|E_3|-x}^3\} \setminus \{l_1\}$$

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3-x}^3, \dots, j_{|E_3|-x}^3\} \setminus \{l_2\}$$

où, $l_1 \in L_1 = \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\}$, $x \in \{1, \dots, q_3 - 1\}$ et $l_2 \in L_2 = \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\}$, ne sont pas des couvertures minimales. D'où, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1} \setminus \{l_1\}} e_j + \sum_{j \in E_{q_2}^{E_{\#}^2}} e_j + \sum_{j \in E_{q_3+1}^{E_{\#}^3}} e_j$$

et les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_{\#}^1}} e_j + \sum_{j \in E_{q_2}^{E_{\#}^2} \setminus \{l_2\}} e_j + \sum_{j \in E_{q_3+1}^{E_{\#}^3}} e_j$$

sont faisables. par conséquent, nous obtenons $q_1 + q_2$ points.

- Par hypothèse, si $E_{\#}^1 - q_1 \geq 1$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_1^1, j_{|E_1|-q_1+3}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3+1}^3, \dots, j_{|E_3|}^3\}$$

n'est pas une couverture, comme l'inégalité (5.16) domine une autre inégalité, alors les points :

$$\sum_{i \in C_p} e_i + e_{l_3} + \sum_{j \in E_{q_1-2}^{E_1^\#}} e_j + \sum_{j \in E_{q_2}^{E_2^\#}} e_j + \sum_{j \in E_{q_3+1}^{E_3^\#}} e_j$$

où, $l_3 = 1, \dots, E_1^\# - q_1$, sont faisables. Donc, nous aurons $E_1^\# - q_1$ points.

- Par hypothèse, si $E_2^\# - q_2 \geq 1$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_1^2, j_{|E_2|-q_2+3}^2, \dots, j_{|E_2|}^2\} \cup \{j_{|E_3|-q_3+1}^3, \dots, j_{|E_3|}^3\}$$

n'est pas une couverture, comme l'inégalité (5.16) domine une autre inégalité, alors les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_1^\#}} e_j + e_{l_4} + \sum_{j \in E_{q_2-2}^{E_2^\#}} e_j + \sum_{j \in E_{q_3+1}^{E_3^\#}} e_j$$

où, $l_4 = 1, \dots, E_2^\# - q_2$, sont faisables. Donc, nous aurons $E_2^\# - q_2$ points.

- Par hypothèse, si $E_3^\# - q_3 \geq 2$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \{j_{|E_2|-q_2+1}^2, \dots, j_{|E_2|}^2\} \cup \{j_1^3, j_{|E_3|-q_3+2}^3, \dots, j_{|E_3|}^3\}$$

n'est pas une couverture. D'où, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_1^\#}} e_j + \sum_{j \in E_{q_2}^{E_2^\#}} e_j + e_{l_5} + \sum_{j \in E_{q_3-1}^{E_3^\#}} e_j$$

où, $l_5 = 1, \dots, E_3^\# - q_3 - 1$, sont faisables. Ainsi, nous obtenons $E_3^\# - q_3 - 1$ points.

- Si $E_3^\# \neq E_3$ et $\beta_{E_2^\#, E_3^\#} > \beta_{E_2^\#, E_3^\#+1}$, alors l'algorithme 3 change la valeur de $\beta_{E_2^\#, E_3^\#}$ pendant la $(p, E_2^\#, E_3^\# + 1)^{\text{ème}}$ itération. Par conséquent, le point :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_1^\#}} e_j + \sum_{j \in E_{q_2}^{E_2^\#}} e_j + \sum_{j \in E_{q_3+1}^{E_3^\#+1}} e_j$$

est faisable. Ainsi, les points :

$$\sum_{i \in C_p} e_i + \sum_{j \in E_{q_1}^{E_1^\#}} e_j + \sum_{j \in E_{q_2}^{E_2^\#}} e_j + \sum_{j \in E_{q_3}^{E_3^\#}} e_j + e_{l_6}$$

sont faisables, pour tout $l_6 \in E_3 \setminus E_3^\#$. Ainsi, nous avons $|E_3 \setminus E_3^\#|$ points.

Il est clair que ces points sont affinement indépendants, car la matrice de ces points comporte des sous matrices cycliquement permutées avec un seul zéro sur ces permutations. De plus, la matrice comporte aussi des lignes qui ont un seul un. \square

5.1.5 Complexité

La complexité de l'algorithme 3 est donnée par le théorème suivant.

Théorème 5.2. *La complexité de l'algorithme 3 est de l'ordre de :*

$$O(|C||E_1||E_2||E_3| + |C|\log|C| + |E_1|\log|E_1| + |E_2|\log|E_2| + |E_3|\log|E_3|)$$

Preuve. Nous avons quatre ensembles principaux, C (une couverture minimale), E_1 , E_2 et E_3 ($E_1 \cup E_2 \cup E_3 = N \setminus C$), ces ensembles sont triés dans un ordre décroissant de leurs coefficients w_j . La complexité de cette procédure (le trie) est de l'ordre de $O(|C|\log|C|)$, $O(|E_1|\log|E_1|)$, $O(|E_2|\log|E_2|)$ et $O(|E_3|\log|E_3|)$, respectivement.

L'algorithme prend un nombre constant p d'indices de la couverture minimale C et itère pour tous les indices : $1, \dots, |E_1|$. Dans chaque iteration et pour chacun de ces indices, nous itérons pour tous les indices : $1, \dots, |E_2|$ de E_2 , et dans chacune de ces dernières iterations et pour chacun des indices de E_2 nous itérons pour tous les indices : $1, \dots, |E_3|$ de E_3 , d'où, ces trois boucles imbriquées sont de l'ordre de $O(|E_1||E_2||E_3|)$. Ces itérations sont alors répétées pour les valeurs variables de p , qui peuvent avoir a nombre maximum $|C|$. Nous pouvons clairement voir que la complexité des quatre boucles imbriquées est de l'ordre de $O(|C||E_1||E_2||E_3|)$.

Ainsi, l'algorithme 3 est de l'ordre de $O(|C||E_1||E_2||E_3| + |C|\log|C| + |E_1|\log|E_1| + |E_2|\log|E_2| + |E_3|\log|E_3|)$. \square

5.2 Lifting simultané de plusieurs ensembles de variables

5.2.1 Présentation de l'algorithme

Pour le lifting simultané de plusieurs ensembles de variables, E_i , où $i = 1..m$, nous allons opérer de la même manière que pour le lifting simultané de deux et trois ensembles de variables et développer un algorithme, algorithme 4, qui permet de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. L'algorithme 4 a le même principe que l'algorithme 2 et 3, i.e. à chaque itération du parcours de l'ensemble E_i nous testons si c'est possible ($T_i \leq d$) de parcourir le $(i+1)^{\text{ème}}$ ensemble E_{i+1} et de prendre un nombre de variables, noté q_{i+1} , pour les lifter. Les inégalités générées par cet algorithme sont de la forme suivante :

$$\sum_{i \in C} x_i + \sum_{l=1}^i \alpha^l \sum_{j \in E_l} x_j \leq |C| - 1 \quad i = 1..m \quad (5.17)$$

où $\alpha^i = \frac{|C|-1-p}{q_i}$ avec $i = 1..m$.

Sachant que les ensembles C et E_i , où $i = 1..m$, sont triés selon l'ordre décroissant de leurs coefficients w_j tel que : $w_i \leq w_{i+1}$, $w_{j^i} \leq w_{j^i+1}$ pour C, E_i , pour $i = 1, \dots, m$, respectivement.

5.2.1.1 Principe de l'algorithme

Nous choisissons p indices de la couverture minimale C correspondant aux plus petit coefficient w_j et nous les fixons à 1, leur somme est donnée par C_Σ . Pour une valeur de i qui varie de 1 à $m-1$, nous choisissons q_i éléments, où $q_i = 1..|E_i|$, de l'ensemble E_i et nous calculons la somme E_Σ^i . Les valeurs de α^i sont initialisées à : $\alpha^1 = \infty$ et $\alpha^i = \alpha^{i-1}$. Le choix des éléments dans les ensembles E_i se fait séquentiellement ; i.e. nous parcourons l'ensemble E_i , nous prenons q_i variables pour les lifter et nous testons si c'est possible ($T_i \leq d$) de parcourir le $(i+1)^{ème}$ ensemble E_{i+1} et prendre q_{i+1} variables pour les lifter. La procedure se résume comme suit :

Nous calculons le total $T_i = C_\Sigma + \sum_{l=1}^i E_\Sigma^l$.

1. Si $T_i \leq d$, alors p éléments de C et q_i éléments de $\{j_1^i, \dots, j_{E_\#}^i\}$ pour les l ensembles E_i , où $l = 1..i$, ne forment pas une couverture, donc, il existe le point faisable correspondant et la valeur du coefficient α^i doit être changée de sorte que ce point avec p et $\sum_{l=1}^i q_l$ éléments satisfait l'inégalité (5.17) à égalité.
2. Si $T_i > d$, alors les p éléments de C et q_i éléments de $\{j_1^i, \dots, j_{E_\#}^i\}$ pour les l ensembles E_i , où $l = 1..i$, forment une couverture, donc, le coefficient du lifting α^i reste inchangé ($\alpha_{E_\#, E_\#}^i = \alpha_{E_\#, E_\#}^{i-1}$).

Cette procedure sera répétée pour tout $p \in \{|C| - 1, \dots, 0\}$. Lorsque nous terminons de dérouler l'algorithme, nous maintenons la valeur de $\alpha_{p, E_\#}^{*1}$ et $\alpha_{E_\#, E_\#}^{*i}$ qui correspondent aux $\text{minimum}_{p \in \{0, 1, \dots, |C|-1\}} \{\alpha_{p, E_\#}^1\}$ et $\text{minimum}_{p \in \{0, 1, \dots, |C|-1\}} \{\alpha_{E_\#, E_\#}^i\}$, ces derniers représentent les coefficients du lifting simultané optimaux de $E_\#^1$ et $\sum_{l=1}^i E_\#^l$ variables respectivement. Cependant, nous pouvons facilement générer les inégalités maximales du lifting simultané en choisissant les inégalités ayant la plus grande valeur de $E_\#^1$ et $E_\#^i$ pour $\alpha_{p, E_\#}^{*1}$ et $\alpha_{E_\#, E_\#}^{*i}$ donnés respectivement.

5.2.2 Calcul des coefficients α^i

Le calcul des coefficients α^i , pour les inégalités de la forme (5.17), se fait de la manière suivante :

Si $T_2 \leq d$, cela implique que $C_\Sigma + \sum_{l=1}^i E_\Sigma^l \leq d$, d'où, p éléments de C et $\sum_{l=1}^i q_l$ éléments de $\{j_1^i, \dots, j_{E_\Sigma^i}^i\}$, où $i = 1..m$, ne forment pas une couverture. Cependant, il existe le point faisable correspondant et la valeur du coefficient α^i doit être changée de sorte que ce point avec p et $\sum_{l=1}^i q_l$ éléments satisfait l'inégalité (5.17) à égalité.

Nous avons :

$$\alpha^i = \frac{|C| - 1 - p}{q_i} \quad (5.18)$$

Par substitution dans l'inégalité (5.17), pour $\sum_{i \in C} x_i = p$, $\sum_{j \in E_i} x_j = q_i$, où $i = 1..m$, nous aurons :

$$\frac{|C| - 1 - p}{q_1} \times q_1 + \dots + \frac{|C| - 1 - p}{q_i} \times q_i = i(|C| - 1 - p) > |C| - 1 - p$$

En multipliant les coefficients α^i par λ_i , nous aurons :

$$\alpha^i = \frac{|C| - 1 - p}{q_i} \times \lambda_i \quad (5.19)$$

où, $\lambda_i \in]0, 1[$ pour $i = 1..m$, et par substitution dans l'inégalité (5.17), nous obtenons :

$$\frac{|C| - 1 - p}{q_1} \times \lambda_1 \times q_1 + \dots + \frac{|C| - 1 - p}{q_i} \times \lambda_i \times q_i = \left(\sum_{l=1}^i \lambda_l \right) (|C| - 1 - p)$$

Pour que l'inégalité (5.17) soit satisfaite à égalité nous devons avoir : $\sum_{l=1}^i \lambda_l = 1$ Nous avons choisi

de prendre : $\lambda_i = \frac{q_i}{\sum_{l=1}^i q_l}$ est le taux des éléments de E_i pris dans $N \setminus C$. D'où :

$$\alpha^i = \frac{|C| - 1 - p}{q_i} \times \frac{q_i}{\sum_{l=1}^i q_l} = \frac{|C| - 1 - p}{\sum_{l=1}^i q_l} \quad \left. \right\} \Rightarrow \alpha^i = \alpha^{i+1} = \frac{|C| - 1 - p}{\sum_{l=1}^i q_l} \quad \text{où, } i=1..m$$

5.2.3 Formes des inégalités générées

Puisque $\alpha^i = \alpha^{i+1}$ alors les inégalités générées par l'algorithme 4 auront la forme suivante :

$$\sum_{i \in C} x_i + \alpha \sum_{l=1}^i \sum_{j \in E_i} x_j \leq |C| - 1 \quad i = 1..m \quad \alpha = \alpha^i = \alpha^{i+1} = \dots \quad (5.20)$$

Algorithme 4 : Algorithme pour le lifting simultané de plusieurs ensembles de variables

Données : Une contrainte du problème du sac à dos : $\sum_{j \in N} w_j x_j \leq d$.

m ensembles de variables $E_i = \{j_1^i, j_2^i, \dots, j_{e_i}^i\}$, où $i = 1..m$,

$e_i = |E_i|$, $m < n$ et $N \setminus C = \bigcup_{i=1}^m E_i$

Résultats : Des inégalités de la forme (5.20)

$p := |C| - 1$;

while $p \geq 0$ **do**

$C_\Sigma := \sum_{r=|C|-p+1}^{|C|} w_{i_r}$; $\alpha_{p,0}^1 := \infty$; $q_1 := 1$; $E_\#^1 := 1$;

while $E_\#^1 \leq |E_1|$ **do**

$E_\Sigma^1 := \sum_{r=E_\#^1-q_1+1}^{E_\#^1} w_{j_r^1}$; $T_1 = C_\Sigma + E_\Sigma^1$;

if $T_1 \leq b$ **then**

$\alpha_{p,E_\#^1}^1 := \frac{|C|-1-p}{q_1}$; $q_1 := q_1 + 1$;

\vdots

$q_m := 1$; $\alpha_{E_\#^{m-1},0}^m := \alpha'_{E_\#^{m-1},E_\#^m}$; $E_\#^m := 1$;

while $E_\#^m \leq |E_m|$ **do**

$E_\Sigma^m := \sum_{r=E_\#^m-q_m+1}^{E_\#^m} w_{j_r^m}$; $T_m = C_\Sigma + E_\Sigma^1 + \dots + E_\Sigma^m$;

if $T_m \leq b$ **then**

$\alpha_{E_{num}^{m-1},E_\#^m}^m := \frac{|C|-1-p}{q_1+\dots+q_m}$; $q_m := q_m + 1$;

else

$\alpha_{E_\#^{m-1},E_\#^m}^m := \alpha_{E_\#^{m-1},E_\#^{m-1}}$;

$E_\#^m := E_\#^m + 1$;

\vdots

else

$\alpha_{p,E_\#^1}^1 := \alpha_{p,E_\#^1-1}^1$;

$E_\#^1 := E_\#^1 + 1$;

$p := p - 1$;

5.2.4 Aspect théorique

Dans le théorème suivant, nous allons établir les conditions nécessaires pour que les inégalités non-dominées générées par l'algorithme 4 soient des facettes pour le polytope du sac à dos.

Théorème 5.3. *Toute inégalité non dominée générée par l'algorithme 4 définit une facette sur $\mathcal{P}_{C \cup (\cup_{i=1}^m E_i)}$ si les conditions suivantes sont satisfaites :*

i) L'inégalité domine une autre inégalité générée par l'algorithme 4;

ii) Si $E_{\#}^i - q_i \geq 1$, alors l'ensemble suivant n'est pas une couverture :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \dots \cup \{j_i^1, j_{|E_i|-q_i+3}^i, \dots, j_{|E_i|}^i\} \cup \dots \cup \{j_{|E_m|-q_m+1}^m, \dots, j_{|E_m|}^m\}$$

où $i=1..m-1$;

iii) Si $E_{\#}^m - q_m \geq 2$, alors l'ensemble suivant n'est pas une couverture :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \{j_{|E_1|-q_1+1}^1, \dots, j_{|E_1|}^1\} \cup \dots \cup \{j_{|E_{m-1}|-q_{m-1}+1}^{m-1}, \dots, j_{|E_{m-1}|}^{m-1}\} \cup \{j_1^m, j_{|E_m|-q_m+2}^m, \dots, j_{|E_m|}^m\}$$

iv) Si $E_{\#}^m \neq E_m$, alors $\alpha_{E_{\#}^m, E_{\#}^m}^m > \alpha_{E_{\#}^m, E_{\#}^m+1}^m$.

Preuve. Nous allons démontrer que les inégalités de la forme (5.20) générées par l'algorithme 4 et qui vérifient les conditions du théorème 5.3 sont des facettes.

Dans cette démonstration nous allons utiliser la technique de preuve directe de facettes, i.e. nous allons montrer que la dimension des faces définies par les inégalités générées par l'algorithme 4 est $\dim(\mathcal{P}) - 1$.

Les inégalités générées par l'algorithme 4 sont valides car ce dernier vérifie chaque point qui peut les rendre invalides.

Il est clair que la dimension de $\text{conv}(\mathcal{P}_{C \cup (\cup_{i=1}^m E_i)})$ est $|C| + (\sum_{i=1}^n |E_i|)$ car l'origine ne satisfait pas toute inégalité de couverture simultanément liftée à égalité. Ainsi, d'après le théorème 2.1, la dimension de la face induite par toute inégalité générée :

$$\sum_{i \in C} x_i + \alpha_{E_{\#}^i, E_{\#}^i+1} (\sum_{i=1}^m \sum_{j \in E_{\#}^i} x_j) \leq |C| - 1 \quad (5.21)$$

générée par l'algorithme 4 peut être au plus $|C| + (\sum_{i=1}^n |E_i|) - 1$.

Les $|C| + (\sum_{i=1}^n |E_i|)$ points affinement indépendants sont déterminés comme suit :

- L'algorithme 4 exige que C soit minimal et ainsi le point $\sum_{i \in C \setminus \{h\}} e_i$, est faisable et satisfait cette inégalité à égalité, pour chaque $h \in C$. D'où, nous avons $|C|$ points.
- Puisque l'inégalité (5.21) domine au moins une inégalité, donc $\alpha_{E_{\#}, E_{\#}^{i+1}-1}^m = \alpha_{E_{\#}, E_{\#}^{i+1}}^m$. Par conséquent, le point :

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_j + \sum_{k \in E_{q_m}^{E_{\#}^{m-1}}} e_k$$

est faisable. Puisque l'ensemble E_m est trié, les points :

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_j + \sum_{k \in E_{q_{m+1}}^{E_{\#}^m} \setminus \{k\}} e_k$$

sont faisables, pour tout $k \in E_{q_{m+1}}^{E_{\#}^m}$. Nous obtenons $q_m + 1$ points.

- De plus si l'inégalité (5.21) domine une autre inégalité, l'ensemble :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \left(\bigcup_{i=1}^{m-1} \{j_{|E_i|-q_i+1}^i, \dots, j_{|E_i|}^i\} \right) \cup \{j_{|E_m|-q_m-x}^m, \dots, j_{|E_m|-x}^m\}$$

est une couverture minimale. D'où, l'ensemble

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \left(\bigcup_{i=1}^{m-1} \{j_{|E_i|-q_i+1}^i, \dots, j_{|E_i|}^i\} \right) \cup \{j_{|E_m|-q_m-x}^m, \dots, j_{|E_m|-x}^m\} \setminus \{l\}$$

où, $l \in L = \{j_{|E_i|-q_i+1}^i, \dots, j_{|E_i|}^i\}$ et $x \in \{1, \dots, q_m - 1\}$, n'est pas une couverture minimale, alors, les points :

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-2} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_{j_i} + \sum_{j \in E_{q_k}^{E_{\#}^k} \setminus \{l\}} e_{j_k} + \sum_{j^m \in E_{q_{m+1}}^{E_{\#}^m}} e_{j^m} \quad k = 1, \dots, m-1$$

sont faisables. Ainsi, nous obtenons q_i points.

- Par hypothèse, si $E_{\#}^i - q_i \geq 1$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \left(\bigcup_{i=1, i \neq k}^{m-2} \{j_{|E_i|-q_i+1}^i, \dots, j_{|E_i|}^i\} \cup \{j_1, j_{|E_k|-q_k+3}^k, \dots, j_{|E_k|}^k\} \right) \cup \{j_{|E_m|-q_{m+1}}^m, \dots, j_{|E_m|}^m\}$$

où $k = 1..m-1$, n'est pas une couverture et puisque l'inégalité (5.21) domine une autre inégalité alors les points :

$$\sum_{i \in C_p} e_i + e_l + \sum_{j \in E_{q_{k-2}}^{E_{\#}^k}} e_{j_k} + \sum_{i=1}^{m-2} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_{j_i} + \sum_{j^m \in E_{q_{m+1}}^{E_{\#}^m}} e_{j^m} \quad k = 1, \dots, m-1$$

où, $l = 1..E_{\#}^k - q_k$, sont faisables. Donc, nous aurons $E_{\#}^k - q_k$ points.

- Par hypothèse, si $E_{\#}^m - q_m \geq 2$, alors :

$$\{i_{|C|-p+1}, \dots, i_{|C|}\} \cup \left(\bigcup_{i=1}^{m-1} \{j_{|E_i|-q_{i+1}}^i, \dots, j_{|E_i|}^i\} \right) \cup \{j_1^m, j_{|E_m|-q_{m+2}}^m, \dots, j_{|E_m|}^m\}$$

n'est pas une couverture. D'où, les points :

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_{j^i} + e_l + \sum_{j^m \in E_{q_{m-1}}^{E_{\#}^m}} e_{j^m}$$

où, $l = 1..E_{\#}^m - q_m - 1$, sont faisables. Ainsi, nous obtenons $E_{\#}^m - q_m - 1$ points.

- Si $E_{\#}^{E_{\#}^m} \neq E_m$ et $\alpha_{E_{\#}^{m-1}, E_{\#}^m}^m > \alpha_{E_{\#}^{m-1}, E_{\#}^{m+1}}^m$, alors l'algorithme 4 change la valeur de $\alpha_{E_{\#}^{m-1}, E_{\#}^m}^m$ pendant la $(p, E_{\#}^{m-1}, E_{\#}^m + 1)$ ^{ème} itération. Par conséquent, le point :

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_{j^i} + \sum_{j^m \in E_{q_{m+1}}^{E_{\#}^{m+1}}} e_{j^m}$$

est faisable. Ainsi, les points :

$$\sum_{i \in C_p} e_i + \sum_{i=1}^{m-1} \sum_{j \in E_{q_i}^{E_{\#}^i}} e_{j^i} + \sum_{j^m \in E_{q_m}^{E_{\#}^m}} e_{j^m}$$

sont faisables pour tout $l \in E_m \setminus E_{\#}^{E_{\#}^m}$. Donc, nous avons $|E_m \setminus E_{\#}^{E_{\#}^m}|$ points.

Il est clair que ces points sont affinement indépendants, car la matrice de ces points comporte des sous matrices cycliquement permutées avec un seul zéro sur ces permutations. De plus, la matrice comporte aussi des lignes qui ont un seul un. \square

5.2.5 Complexité

La complexité de l'algorithme 4 est donnée par le théorème suivant.

Théorème 5.4. *La complexité de l'algorithme 3 est de l'ordre de :*

$$O(|C| \prod_{i=1}^m |E_i| + |C| \log |C| + \sum_{i=1}^m |E_i| \log |E_i|)$$

Preuve. Nous avons $m+1$ ensembles principaux, C (une couverture minimale) et $E_i, i = 1..m$ ($\bigcup_{i=1}^m = N \setminus C$), ces ensembles sont triés dans un ordre décroissant de leurs coefficients w_j . La complexité de cette procédure (le trie) est de l'ordre de $O(|C| \log |C|)$ et $|E_i| \log |E_i|, i = 1..m$, respectivement.

L'algorithme prend un nombre constant p d'indices de la couverture minimale C et itère pour tous les indices : $1..|E_i|$ de E_i , où $i = 1..m$. Dans chaque iteration et pour chacun de ces indices l'algorithme itère pour tous les indices dans E_{i+1} , où $i = 2..m$, d'où, ces m boucles imbriquées sont de l'ordre de $O(\prod_{i=1}^m |E_i|)$. Ces itérations sont alors répétées pour les valeurs variables de p , qui peuvent avoir un nombre maximum de $|C|$. Nous pouvons clairement voir que la complexité des m boucles imbriquées est de l'ordre de $O(|C| \prod_{i=1}^m |E_i|)$.

Ainsi, l'algorithme 4 est de l'ordre de $O(|C| \prod_{i=1}^m |E_i| + |C| \log |C| + \sum_{i=1}^m |E_i| \log |E_i|)$. \square

Conclusion

Dans ce chapitre nous avons développé un algorithme, algorithme 3, qui permet de lifter simultanément trois ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Ensuite, nous avons proposé une généralisation du lifting simultané à plusieurs ensembles de variables en développant un autre algorithme, algorithme 4, qui permet de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Pour chacun de ces algorithmes, nous avons énoncé les conditions nécessaires pour que les inégalités non-dominées qu'ils générèrent soient des facettes pour \mathcal{P} .

Conclusion générale

Dans cette thèse, notre intérêt a porté sur l'une des approches les plus utilisées pour la résolution des problèmes d'optimisation combinatoire, à savoir l'approche polyédrale. Plus précisément, nous avons traité un cas particulier d'extension d'inégalités valides : la technique de lifting. Cette technique permet d'étendre une inégalité valide en introduisant plus de variables dans l'inégalité. Nous distinguons, selon la manière dont les variables sont introduites, le lifting séquentiel et le lifting simultané.

Le lifting simultané fait l'objet de la contribution de notre travail qui a pour objectif de développer des algorithmes qui permettent de lifter simultanément plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Ce dernier est l'un des problèmes *NP*-complet qui présente un très grand intérêt que ce soit dans le domaine pratique ou théorique.

Dans un premier temps, nous avons développé un algorithme (algorithme 2) qui permet de lifter simultanément deux ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Ensuite, nous avons énoncé les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du sac à dos. Puis, nous avons effectué une comparaison entre cet algorithme et MSLA basée sur le déroulement d'un exemple.

Par la suite, nous avons essayé de généraliser le lifting simultané sur plusieurs ensembles de variables. D'abord, nous avons commencé par développer un algorithme (algorithme 3) qui permet de lifter simultanément trois ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire et énoncé les conditions nécessaires pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du sac à dos. Finalement, nous avons proposé une généralisation du lifting simultané sur plusieurs ensembles de variables en développant un autre algorithme (algorithme 4) et énoncé les conditions nécessaires

pour que les inégalités non-dominées générées par cet algorithme soient des facettes pour le polytope du sac à dos.

Comme perspectives de ce travail nous proposons :

- Une étude concernant la taille et le nombre d'ensembles à lifter simultanément en une inégalité de couverture pour le problème du sac à dos binaire.
- D'effectuer une comparaisons entre ces algorithmes et MSLA.
- Une étude sur les valeurs que peut prendre les paramètres λ_i dans l'intervalle $[0, 1]$, pour pouvoir tirer plus d'inégalités.
- Une étude expérimentale avec le logiciel CPLEX pour le travail présenté dans cette thèse et pour les travaux proposés en perspectives.

BIBLIOGRAPHIE

- [1] K. Aardal and C. P. M. Hoesel. Polyhedral techniques in combinatorial optimization i : Theory. Technical report, Utrecht University, 1995.
- [2] K. Aardal and C. P. M. Hoesel. Polyhedral techniques in combinatorial optimization ii : applications and computations. *Statistica Neerlandica*, 53(2) :131–177, 1999.
- [3] K. Aardal and R. Weismantel. Polyhedral combinatorics : An annotated bibliography. In *Annotated Bibliographies in Combinatorial Optimization*, chapter 3. Wiley, 1997.
- [4] A. Bachem and M. Grötschel. New aspect of polyhedral theory. In B. Korte, editor, *Optimization and Operations Research*, Modern Applied Mathematics, chapter 2, pages 51–106. North Holland Publishing Company, 1982.
- [5] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8 :146–164, 1975.
- [6] E. Balas and E. Zemel. Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, 34 :119–148, 1978.
- [7] E. Balas and E. Zemel. An algorithm for large zero-one knapsack problems. *Operations Research*, 28(5) :1130–1154, October 1980.
- [8] E. Balas and E. Zemel. Lifting and complementing yields all facets of positive zero-one programming polytopes. *Proceedings of the International Conference on Mathematical Programming (R.W. Cottle et. al., ed.)*, pages 13–24, 1984.
- [9] M. Baïou. Polyhedral approach to solve combinatorial optimization problems. Laboratoire LIMOS, Université Clermont II, 2007.
- [10] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch and price : Column generation for solving huge integer programs. *Operations Research*, 46 :316–329, 1998.
- [11] R. Bellman. *Dynamic programming*. Princeton NJ, 1957.

-
- [12] R. Bellman and S.E. Dreyfus. Applied dynamic programming. *Princeton University Press*, Princeton NJ, 1962.
- [13] M. P. Bonami. *Étude et mise en oeuvre d'approches polyédriques pour la résolution de programmes en nombres entiers ou mixtes généraux*. PhD thesis, L'université Paris 6, 2003.
- [14] P. Bonami and M. Minoux. Une comparaison de quelques méthodes de génération d'inégalités valides pour les problèmes entiers 0-1 généraux. In *Optimisation Combinatoire 3 : Applications*, Informatique et Systèmes d'Information, chapter 5, pages 169–178. Lavoisier, 2006.
- [15] D. Bouali. *Heuristiques et Approche Polyédrale du Problème de Voyageur de Commerce International*. PhD thesis, Institut National Polytechnique de Grenoble, 1996.
- [16] S. Boyd, A. Ghosh, and A. Magnani. Branch and bound methods. EE392o, November 2003.
- [17] I. Charon and O. Hudry. Méthodes arborescentes par séparation et évaluation (branch and bound). In *Optimisation Combinatoire 1 : Concepts Fondamentaux*, Informatique et Systèmes d'Information, chapter 3, pages 63–94. Lavoisier, 2006.
- [18] V. Chvátal. Hard knapsack problems. *Operations Research*, 28(6) :1402–1411, December 1980.
- [19] V. Chvátal. *Linear Programming*. W.H Freeman and Company, 1983.
- [20] J. Clausen. Branch and bound algorithms : Principles and examples. Technical report, University of Copenhagen, 1999.
- [21] S. Cook. The complexity of theorem-proving procedures. *ACM Press*, STOC() :151–158, 1971.
- [22] J. Cord. A method for allocating funds to investment projects when returns are subject to uncertainty. *Management Science*, 10(2) :335–341, 1964.
- [23] H. Crowder, E.L. Johnson, and M. Padberg. Solving large-scale zero-one linear programming problems. *Operations research*, 31(5) :803–834, October 1983.
- [24] G. Dahl. An introduction to convexity, polyhedral theory and combinatorial optimization. Technical report, University of Oslo, Department of Informatics, 1997.
- [25] R.J. Dakin. A tree search algorithm for mixed integer programming problems. *Computer Journal*, 8 :250–255, 1965.
- [26] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, New Jersey, 1963.

-
- [27] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of large-scale traveling salesman problem. *Journal of Operations Research Society of America*, 2 :393–410, 1954.
- [28] G.B. Dantzig. Discrete variables extremum problems. *Operations Research*, 5 :266–277, 1957.
- [29] J. Delorme and E. Heurigon. Problèmes de partitionnement : exploration arborescente ou méthodes de troncatures? *R.A.I.R.O*, 2(9ème année) :53–65, 1975.
- [30] W. Diffe and M.E. Hellman. New direction in cryptography. *IEEE Transactions Information Theory*, IT-36 :644–654S, 1976.
- [31] K. Dudzinski and S. Walukiewicz. Upper bounds for the 0 – 1 knapsack problem. Report MPD-10-49/84, Systems Research Institute, 1984.
- [32] T. Easton and K. Hooker. Simultaneously lifting sets of binary variables into cover inequalities for knapsack polytopes. *Discrete Optimization*, May 2007.
- [33] J. Edmonds. Maximum matching and a polyhedron with 0 – 1 vertices. *Journal of Research National Bureau of Standards*, 69(B) :125–130, 1965.
- [34] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1 :185–204, 1977.
- [35] D. Fayard and G. Plateau. An algorithm for solution of the 0 – 1 knapsack problem. *Computing*, 28 :269–287, 1982.
- [36] C. Fricke. Examples of new facets for the precedence constrained knapsack problem. Technical report, Department of Mathematics and Statistics, University of Melbourne, 2006.
- [37] A. Fréville. Méthodes de recherche locale. In *Journée AFPLC*. Ecole des Mines de Nantes, 2000.
- [38] P.C. Gilmore and R.E. Gomory. The theory and computation of knapsack functions. *Operations Research*, 14 :1045–1074, 1966.
- [39] Z. Gu G.L. Nemhauser and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs : computation. *INFORMS Journal on Computing*, 10(4) :427–437, 1998.
- [40] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull Am Math Soc*, 64 :275–278, 1958.
- [41] R. E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications*, 2 :451–588, 1969.

-
- [42] H. Greenberg and R. L. Hegerich. a branch search algorithm for the knapsack problem. *Management Science*, 16 :327–332, 1970.
- [43] M. Grötschel. Polyhedral combinatorics. Technical report, Augsburg University, 1984.
- [44] M. Grötschel and L. Lovász. Combinatorial optimization : A survey. Technical Report 29, DIAMACS Technical Report, 1993.
- [45] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1 :70–89, 1981.
- [46] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, New York, 1988.
- [47] M. Grötschel and M.W. Padberg. Polyhedral computation. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B.Shmoys, editors, *The Traveling Salesman Problem*, chapter 9, pages 307–360. John Wiley & Sons, 1985.
- [48] M. Grötschel and M.W. Padberg. Polyhedral theory. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B.Shmoys, editors, *The Traveling Salesman Problem*, chapter 8, pages 251–305. John Wiley & Sons, 1985.
- [49] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs : complexity. *Mathematecal Programming*, 85(3) :439–467, 1999.
- [50] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted flow cover inequalities for mixed 0-1 integer programs. *Mathematecal Programming*, 85(Series A) :436–467, 1999.
- [51] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Sequence independent lifting in mixed integer programming. *Combinatorial Optimization*, 4 :109–129, 2000.
- [52] S.R. Gupta, R.L. Bulfin, and J.S. Smith. Lifting cover inequalities for the binary knapsack polytope. Technical report, Industrial and Systems Engineering Department, Auburn University, 2006.
- [53] Gutierrez. *Lifting general integer variables*. PhD thesis, Kansas State University, MS Dissertation, Department of Industrial and Manufacturing Systems Engineering, 2007.
- [54] P. L. Hammer, E. L. Johnson, and U. N. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8 :179–206, 1975.
- [55] F. Hansmann. Operations research in the national planning of under developed countries. *Operations Research*, 9 :203–248, 1961.

-
- [56] D. Hartvigsen and E. Zemel. The complexity of lifted inequalities for the knapsack problem. *Discrete Applied Mathematics*, 39 :113–123, 1992.
- [57] A. J. Hoffman and J. Kruskal. Integer boundary points of convex polyhedra. In Princeton University Press, editor, *Linear Inequalities and related Systems*, pages 223–246. H. W. Kuhn and A. W. Tucker, Princeton, 1956.
- [58] E. Horowitz and S. Sahni. Computing partitions with applications to the knapsack problem. *Journal of ACM*, 21 :277–292, 1974.
- [59] O.H. Ibarra and C.E. Kim. Fast approximation algorithms for the knapsack problem and sum of subset problems. *Journal of ACM*, 22 :463–468, 1975.
- [60] G.P. Ingargiola and J.F. Korsh. A reduction algorithm for zero one single knapsack problem. *Management Science*, 20 :460–463, 1973.
- [61] D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Science*, 9 :256–278, 1974.
- [62] K. Kaparis and A.N. Letchford. Separation algorithms for 0-1 knapsack polytopes. Technical report, Lancaster University, 2007.
- [63] S. Kaplan. Solution of the lorie-savage and similar integer programming problems by the generalized lagrange multiplier method. *Operations Research*, 14(6) :1130–1136, 1966.
- [64] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- [65] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer-Verlag, 2003.
- [66] P. J. Kolesar. A branch and bound algorithm for the knapsack problem. *Management Science*, 13 :723–735, 1967.
- [67] B. Korte and J. Vygen. *Combinatorial Optimization : Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, 3 edition, 2005.
- [68] M. G. Lagoudakis. The 0-1 knapsack problem an introductory survey. Technical report, The Center for Advanced Computer Studies, University of Southwestern Louisiana, 1994.
- [69] A.H. Land and A.G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28(497-520), 1960.

-
- [70] E. L. Lawler. Fast approximation algorithms for knapsack problems. *mathematics of Operations Research*, 4 :339–356, 1979.
- [71] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel. An algorithm for the traveling salesman problem. *Operations Research*, 11(6) :972–989, 1963.
- [72] M. J. Magazine and O. Oguz. A fully polynomial approximate algorithm for 0 – 1 knapsack problem. *European Journal of Operations Research*, 8 :270–273, 1981.
- [73] A.R. Mahdjoub. Approches polyédrales. In *Optimisation Combinatoire 1 : Concepts Fondamentaux*, Informatique et Systèmes d’Information, chapter 9, pages 179–202. Lavoisier, 2006.
- [74] A.R. Mahjoub. Efficacité des approches polyédrales en optimisation combinatoire. LIMOS, Université Blaise Pascal, Clermont Ferrand, 2005. Journées Franciliennes de Recherche Opérationnelle, CNAM.
- [75] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123 :397–446, 2002.
- [76] S. Martello and P. Toth. An upper bound for the zero one knapsack problem and branch and bound algorithm. *European Journal of Operational Research*, 1 :169–175, 1977.
- [77] S. Martello and P. Toth. A new algorithm for 0 – 1 knapsack problem. *Management Science*, 34 :633–644, 1988.
- [78] S. Martello and P. Toth. *Knapsack Problems : Algorithms and Computer Implementations*. John Wiley & Sons, Chichester, 1990.
- [79] S. Martello and P. Toth. Upper bounds and algorithms for hard 0 – 1 knapsack problems. *Operations Research*, 45 :768–778, 1997.
- [80] G.B. Mathews. On the partition of numbers. *Proceedings of the London Mathematical Society*, 28 :486–490, 1897.
- [81] J.E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. In *Handbook of Applied Optimization*. Oxford University Press, 2000.
- [82] H. Müller and Merbah. An improved upper bound for zero one knapsack problem : a note on the paper by martello and toth. *European Journal of Operations Research*, 2 :212–213, 1978.
- [83] G. L. Nemhauser and P. H. Vance. Lifted cover facets of the 0-1 knapsack polytope with gub constraints. *Operations Research Letters*, 16(5) :255–263, 1994.

-
- [84] G.L Nemhauser and L.A Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1999.
- [85] M. W. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33 :60–100, 1991.
- [86] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 1 :199–215, 1973.
- [87] M.W. Padberg. (1,k)configurations and facets for packing problems. *Mathematecal Programming*, 18(1) :94–99, Jannuary 1980.
- [88] D. Pisinger. *Algorithms for Knapsack Problems*. PhD thesis, University of Copenhagen, 1995.
- [89] D. Pisinger. The quadratic knapsack problem : a survey. *Discrete Applied Mathematics*, 155(5) :623–648, 2007.
- [90] D. Pisinger and P. Toth. Knapsack problems. In D.-Z. Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 1, pages 299–428. Kluwer Academic Publishers, 1998.
- [91] T. Roy and L. Wolsey. Solving mixed integer programming problems using automatic reformulation. *Operations Research*, 35 :45–57, 1987.
- [92] S. Sahni. Approximate algorithms for the 0 – 1 knapsack problem. *Journal of ACM*, pages 115–124, 22.
- [93] H. M. Salkin and C. A. De Kluyver. The knapsack problem : A survey. *Naval Research Logistics Quarterly*, 22(1) :127–144, 2006.
- [94] H.M. Salkin. Integer programming. *Addison-Wesley Publishing Company*, Inc, 1975.
- [95] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer, 2003.
- [96] A. Schrijver. On the history of combinatorial optimization (till 1960). In K. Aardal, G.L. Nemhauser, and R. Weismantel, editors, *Discrete Optimization*, volume 12 of *Handbooks of Operation Research Managenent Science*, chapter 1, pages 1–68. Elsevier, New York, 2005.
- [97] A. Schrijver. A course in combinatorial optimization. Technical report, Universit  d’Amsterdam, CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands, August 2007.
- [98] K. Sharma. *Simultaneously lifting sets of varibales in binary knapsack problems*. Master of science, Kansas State University, 2007.

-
- [99] A. Sinha and A.A. Zottner. The multiple choice knapsack problem. *Operations Research*, 27 :503–515, 1979.
- [100] C. Solnon. *Contributions à la résolution pratique de problèmes combinatoires des fourmis et des graphes*. PhD thesis, Université Claude Bernard Lyon 1, 2005.
- [101] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time : preliminary report. *Assoc. Comput. Mach.*, Fifth Annual ACM Symposium on Theory of Computing :1–9, 1973.
- [102] M. Syslo, N. Deo, and J.S. Kowalik. Discrete optimization algorithms with pascal programs. *Prentice-Hall, Inc*(Englewood Cliffs), 1983.
- [103] P. Toth. Dynamic programming algorithms for zero one knapsack problem. *Computing*, 25 :29–45, 1980.
- [104] V.V. Vazirani. *Algorithmes d'approximation*. Springer, 2006.
- [105] R. Weismantel. *Knapsack Problems, Test Sets and Polyhedra*. Habilitationsschrift, TU Berlin, Juni 1995.
- [106] R. Weismantel. On the 0/1 knapsack polytope. *Mathematical Programming*, 77(1) :49–68, April 1997.
- [107] L. Wolsey. Faces for a linear inequality in 0-1 variables. *Mathematical Programming*, 8 :165–178, 1975.
- [108] L. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24 :367–372, 1975.
- [109] L.A. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24 :367–372, 1976.
- [110] L.A. Wolsey. Valid inequalities and superadditivity for 0-1 integer programs. *Mathematical Programming*, 2(1) :66–77, February 1977.
- [111] G. Yu. On max min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44(2) :407–415, April 1996.
- [112] E. Zemel. Lifting the facets of zero one polytopes. *Mathematical programming*, 15 :268–277, 1978.
- [113] E. Zemel. The linear multiple choice knapsack problem. *Operations Research*, 28(6) :1412–1423, December 1980.

- [114] E. Zemel. On the computational complexity of facets of the knapsack problem. Technical report, Northwestern University, 1986.
- [115] E. Zemel. Easily computable facets of the knapsack polytope. *Mathematic of Operations Research*, 14(4) :760–766, November 1989.
- [116] B. Zeng and J.P.P. Richard. Sequence independent lifting for 0-1 knapsack problems with disjoint cardinality constraints. Technical report, School of Industrial Engineering, Purdue University, 2006.

Résumé

Dans cette thèse, nous nous intéressons à l'approche polyédrale pour le problème du sac à dos binaire. Nous présentons une technique, très utilisée pour l'extension d'inégalités valides, à savoir le *lifting*, qui consiste à étendre une inégalité valide en y introduisant séquentiellement ou simultanément des variables qu'elle ne contient pas. D'où, nous pouvons distinguer deux types de *lifting* : le *lifting* séquentiel et le *lifting* simultané. L'inégalité de couverture induit une face pour le polytope du sac à dos. Néanmoins, elle peut être *liftée* pour définir une facette. Les inégalités *liftées* de couverture sont avérées très utiles comme plans coupants pour la résolution des programmes linéaire en nombres entiers.

Notre travail traite le développement d'algorithmes polynômiaux pour le *lifting* simultané de plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Notre objectif est de proposer une généralisation du *lifting* simultané pour plusieurs ensembles de variables en une inégalité de couverture pour le problème du sac à dos binaire. Les conditions nécessaires pour que les inégalités non dominées, générées par nos algorithmes, soient des facettes pour le polytope du sac à dos sont présentées.

Mots clé : *Lifting*, Le problème du sac à dos, Approche polyédrale.

Abstract

In this thesis, we are interested in the polyhedral approach for the binary knapsack problem. We present a useful technic, for the extension a valid inequalities, mainly the *lifting*, which consists to spread a valid inequality by introducing sequentially or simultaneously variables that it does not contain. Thus, two kinds of *lifting* can be distinguished : the sequential and the simultaneous *lifting*. The cover inequality induces a face for the knapsack polytope. Nevertheless, it can be *lifted* to define a facet. *Lifted* cover inequalities often prove useful as cutting planes for solving linear integer programming problems.

This work deals with developing polynomial time algorithms for simultaneously *lifting* sets of variables into a cover inequality for the binary knapsack polytope. Our aim is to suggest a generalization of the simultaneous *lifting* on several sets of variables into a cover inequality for the binary knapsack polytope. Necessary conditions for the non dominated inequalities, generated by our algorithms, to define facets for the knapsack polytope are presented.

Keywords : *Lifting*, Knapsack problem, Polyhedral approach.