

## Chapitre 5

# Optimisation de fonctions pseudo-booléennes

Laurent CARAFFA (Ifsttar), Jean-Philippe TAREL (Ifsttar), Mathias PAGET (Ifsttar)

*Une approche pour faire du traitement d'image consiste à poser les problèmes comme la minimisation d'une énergie sur l'espace des images qui sont représentées par des fonctions 2D. L'optimisation de ce type d'énergie passe par le développement de schémas numériques et donc par la nécessaire discrétisation de l'espace des fonctions choisies et de l'énergie utilisée. Les images étant en pratique représentées de façon discrétisée, une autre approche consiste à poser les problèmes comme la minimisation d'une énergie directement dans le domaine discret. Cela conduit généralement à introduire une représentation de l'image sous la forme d'un graphe afin de pouvoir modéliser les interactions entre voisins. Avec cette approche, on dispose du cadre théorique de l'optimisation quadratique pseudo-booléenne (QPBO) dans le cas où les variables sont binaires et de certaines extensions aux cas non-binaires. Dans ce chapitre, les principaux résultats obtenus dans ce cadre théorique QPBO sont présentés de façon succincte afin d'introduire les outils disponibles actuellement. Enfin, l'utilisation de ces outils est illustrée sur le problème de la reconstruction 3D à partir de paires stéréoscopiques, mais ils peuvent s'appliquer à bien d'autres problèmes au delà du traitement d'image.*

### 5.1 Introduction

En traitement d'images et dans d'autres domaines, de nombreux problèmes conduisent à la recherche d'un optimum global d'une énergie multi-labels. La solution recherchée est alors représentée par un ensemble de variables indexées par les sommets d'un graphe et prenant leurs valeurs parmi un ensemble de labels (entiers). Soit  $L = \{0, \dots, m\}$  l'ensemble des labels, par exemple  $\{0, 1\}$

Optimisation de formes en sciences de l'ingénieur

pour une segmentation binaire,  $\{0, \dots, 255\}$  pour la restauration d'une image 8 bits,  $\{0, \dots, m\}$  pour un problème de reconstruction 3D à partir de paires stéréoscopiques avec  $m$  la disparité maximale admissible. Dans ces exemples, le graphe associé a pour sommets les pixels et les arrêtes codent les voisinages entre pixels.

Soit  $n$  le nombre de variables et de sommets,  $f : L^n \rightarrow \mathbb{R}$  est une fonction déduite du modèle du problème à traiter, où chaque configuration  $\mathbf{x}$  de  $L^n$  correspond une valeur d'énergie :

$$f(\mathbf{x}) = \sum_{u \in C_1} \vartheta_u(L_u) + \sum_{u,v \in C_2} \vartheta_{uv}(L_u, L_v) + \sum_{u,v,w \in C_3} \vartheta_{uvw}(L_u, L_v, L_w) + \dots \quad (5.1)$$

où  $C_k$  est l'ensemble des sous-ensembles de  $k$  sommets et  $\vartheta$  est la fonction de coût de chaque sous-ensemble. Lorsque le modèle dérive d'un modèle probabiliste, les solutions les plus probables ont les énergies les plus faibles, on essaiera donc de résoudre le problème de minimisation suivant :

$$\min_{\mathbf{x} \in L^n} f(\mathbf{x}). \quad (5.2)$$

La recherche des labels qui minimisent (5.1) est généralement complexe. Le développement de modèles nécessite donc la connaissance des techniques d'optimisation disponibles.

En optimisation, il est primordial de savoir s'il est possible de trouver, et dans quels délais, le minimum global d'une fonction. Si on peut trouver le minimum global d'une énergie, il n'y a pas d'ambiguïté sur le résultat de l'optimisation, et l'on peut évaluer si le modèle proposé est bien en adéquation avec le problème à résoudre. Inversement, lorsque seulement un minimum local est obtenu, on ne sait pas si les erreurs viennent de ce que l'on n'a pas atteint le minimum global de la fonction, ou si c'est le modèle qui est incorrect.

Lorsque le nombre de configurations possibles n'est pas important, il est assez simple de trouver le minimum de l'énergie. Une recherche exhaustive sur l'ensemble des configurations est suffisante pour permettre de trouver celle qui minimise l'énergie. Mais quand il existe des interactions locales entre les variables, le nombre de configurations possibles devient très important avec le nombre de variables. Par exemple, considérons le cas de l'optimisation d'un champ de Markov binaire sur une image  $512 \times 512$  avec prise en compte des voisinages deux à deux ; il existe alors  $2^{512 \times 512}$  configurations possibles. Sachant que les images actuelles sont constituées de plusieurs millions de pixels et que la cardinalité de l'ensemble des labels peut être beaucoup plus importante (256 pour du débruitage, ou encore plus pour la stéréovision avec une grande base), il est donc de toute évidence impossible de parcourir l'ensemble des solutions. Des algorithmes spécialisés ont donc dû être conçus pour optimiser ce type de problème avec un grand espace de recherche.

En optimisation continue, lorsqu'une fonction est convexe, tous les minima locaux sont globaux. De façon semblable à l'optimisation continue, il existe en

optimisation discrète une classe de fonctions qui permet, malgré une croissance exponentielle de l'espace de recherche, de trouver un minimum global en un temps polynomial. Ce sont les fonctions dites *sous-modulaires*, qu'il est important de distinguer des fonctions non sous-modulaires. Une des méthodes les plus utilisées pour optimiser les fonctions sous-modulaires est connue sous le nom de coupe de graphe (*graph-cuts*), voir par exemple (KOLMOGOROV et ZABIH, 2002).

Toutefois, l'espace des fonctions sous-modulaires est assez restreint. De plus, au delà d'une certaine cardinalité des tailles des voisinages, pouvoir dire d'une fonction qu'elle est sous-modulaire s'avère être un problème *NP-Difficile*. Il arrive aussi fréquemment que l'on veuille optimiser des fonctions non sous-modulaires, il est donc utile dans ce cas, de disposer de méthodes spécifiques d'optimisation même si les résultats obtenus sont partiels. Compte tenu de l'importance de ce type d'optimisation dans différents champs d'application, de nombreux algorithmes ont vu le jour.

L'une des approches les plus efficaces pour optimiser le type d'énergies (5.2) est celle fondée sur l'optimisation de fonctions pseudo-booléennes. L'étude de l'optimisation des fonctions pseudo-booléennes date des années 50. Observées initialement dans la théorie des jeux, ce fut l'une des principales motivations de leur étude en recherche opérationnelle. Par la suite, la découverte de la présence récurrente de fonctions pseudo-booléennes dans un spectre très large d'applications et de domaines a fortement influé sur l'évolution et le gain d'intérêt de ce domaine.

Dans le cas des labels binaires, on nomme  $\mathbf{B} = \{0, 1\}$  l'ensemble des labels. Une fonction  $f : \mathbf{B}^n \rightarrow \mathbb{R}$  est dite pseudo-booléenne. Elle peut être écrite de façon unique comme un polynôme de  $n$  variables de la façon suivante :

$$f(x_1, \dots, x_n) = c_0 + \sum_{i=1}^n c_i x_i + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j + \sum_{1 \leq i < j < k \leq n} c_{ijk} x_i x_j x_k + \dots \quad (5.3)$$

Dans le cas binaire, l'énergie (5.1) peut être réécrite comme une fonction pseudo-booléenne, voir la partie 5.2.2. Dans le cas non binaire, le passage à des fonctions pseudo-booléennes est décrit dans la partie 5.3.2.

Nous avons choisi d'introduire dans la partie 5.2 l'optimisation de fonctions pseudo-booléennes, en suivant la présentation faite dans (BOROS et Peter L HAMMER, 2002). Cette partie est donc une synthèse partielle de cet article d'introduction sur l'état de l'art de l'optimisation de fonctions pseudo-booléennes. L'ensemble des théorèmes que nous présentons est limité à une partie de cet article. Certaines notions sont détaillées par des exemples complémentaires à ceux de l'article. Cette présentation est assez différente de celle proposée habituellement en coupe de graphe, même si on retrouve comme un cas particulier la coupe de graphe habituelle comme expliqué en partie 5.2.7. La partie suivante 5.3 présente des algorithmes pour l'optimisation approchée de fonctions multi-labels fondées sur la théorie de l'optimisation de fonctions pseudo-booléennes. Enfin, dans la partie 5.4, trois exemples d'utilisation de l'optimisation de fonctions pseudo-booléennes sont discutés afin d'expliquer comment utiliser en pratique ce type d'optimisation.

## 5.2 Optimisation de fonctions pseudo-booléennes

### 5.2.1 Notations

Dans la suite, on nomme  $\mathbf{V} = \{1, 2, \dots, n\}$  l'ensemble des indices de 1 à  $n$ , où  $n$  est le nombre de variables.  $\mathbf{x} = (x_1, \dots, x_n)$  est un vecteur binaire et par définition  $\bar{x}_i = 1 - x_i$  est le complément pour  $i \in \mathbf{V}$ . On note  $\mathbf{X} = \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$  l'ensemble de ces symboles.

### 5.2.2 Représentations

On dispose d'au moins deux représentations pour la même fonction pseudo-booléenne.

#### 5.2.2.1 Forme multilinéaire polynomiale unique

$$f(\mathbf{x}) = \sum_{S \subseteq \mathbf{V}} c_S \prod_{j \in S} x_j \quad (5.4)$$

On appelle degré de  $f$  la taille du plus grand ensemble  $S \subseteq \mathbf{V}$  pour lequel  $c_S \neq 0$ . Cette représentation est utile pour étudier certaines propriétés de  $f$ .

#### 5.2.2.2 Posiforme

Une fonction pseudo-booléenne peut toujours être représentée par un polynôme à coefficients positifs, au terme constant près, ou *posiforme* de la forme suivante :

$$\phi(\mathbf{x}) = a_0 + \sum_{T \subseteq \mathbf{X}} a_T \prod_{u \in T} u \quad (5.5)$$

où  $a_T \geq 0$ . Pour représenter une fonction pseudo-booléenne par une posiforme, une façon de procéder consiste à remplacer le premier élément  $x$  par  $1 - \bar{x}$  dans chaque terme où  $c_S < 0$  dans la forme multilinéaire unique. Cette dernière opération est nommée complémentation. Il existe plusieurs posiformes pour une même fonction pseudo-booléenne, donc pour une même table de vérité. On notera  $\mathcal{P}(f)$  la famille des posiformes représentant la même fonction  $f$ .

**Exemple 5.1.** Ces deux posiformes :

$$\phi_{11}(\mathbf{x}) = 5x_1 + 4\bar{x}_1\bar{x}_2x_3 + 7x_1x_2x_4 + 9x_3\bar{x}_4$$

$$\phi_{12}(\mathbf{x}) = x_1 + 4x_1x_2 + 4x_1\bar{x}_2\bar{x}_3 + 7x_1x_2x_4 + 4\bar{x}_2x_3 + 9x_3\bar{x}_4$$

ont la même forme multilinéaire polynomiale unique :

$$f_1(\mathbf{x}) = 5x_1 + 13x_3 - 4x_1x_3 - 4x_2x_3 - 9x_3x_4 + 4x_1x_2x_3 + 7x_1x_2x_4.$$

**TABLE 5.1**

Table de vérité de la fonction pseudo-booléenne de l'exemple 5.1. Les minimums de  $f_1$  sont en gras.

$\mathbf{x}$	<b>0 0</b> 0 0 <b>0 0</b> 0 0 1 1 1 1 1 1 1
	<b>0 0</b> 0 0 1 1 1 1 0 0 0 0 1 1 1
	<b>0 0</b> 1 1 <b>0 0</b> 1 1 0 0 1 1 0 0 1 1
	<b>0 1</b> 0 1 <b>0 1</b> 0 1 0 1 0 1 0 1 0 1
$f_1(\mathbf{x})$	<b>0 0</b> 13 4 <b>0 0</b> 9 <b>0</b> 5 5 14 5 5 12 14 12

En effet, ces trois fonctions pseudo-booléennes vérifient la même table de vérité, table. 5.1.

### 5.2.3 Propriétés des fonctions pseudo-booléennes

La structure de posiforme permet de vérifier certaines propriétés. En effet, une posiforme ne possédant aucun terme négatif, la valeur de la posiforme ne peut être inférieure à 0, à la constante près. Dans un sens, minimiser une posiforme équivaut à essayer d'annuler le plus de termes possible. Il en découle que minimiser une posiforme est équivalent au problème d'optimisation de satisfabilité booléenne maximale *MAX-SAT* (Maximum Satisfiability problem), qui est *NP-complet*. Sous l'hypothèse  $P \neq NP$ , on est alors seulement capable de trouver en un temps polynomial une affectation partielle des variables binaires qui est dans l'*optimum* ((BOROS et Peter L HAMMER, 2002), p.18).

On appelle affectation partielle, un vecteur binaire  $\mathbf{y} \in \mathbf{B}^S$  correspondant à un sous-ensemble  $S \subseteq \mathbf{V}$ . De plus, pour un sous-ensemble  $S \subseteq \mathbf{V}$  d'indices et un vecteur  $\mathbf{y} \in \mathbf{B}^S$ , on note par  $\mathbf{x}[S] \in \mathbf{B}^S$  le vecteur correspondant aux indices dans  $S$ , c'est-à-dire  $\mathbf{x}[S] = (x_i | i \in S)$ . Pour une affectation partielle  $\mathbf{y} \in \mathbf{B}^S$  et pour un vecteur  $\mathbf{x} \in \mathbf{B}^n$ , on définit l'échange de  $\mathbf{x}$  comme le vecteur binaire  $\mathbf{z}$  par :

$$z_j = \begin{cases} x_j & \text{si } j \notin S \\ y_j & \text{si } j \in S \end{cases} \quad (5.6)$$

et on le note par  $\mathbf{z} = \mathbf{x}[S \leftarrow \mathbf{y}]$ . Par exemple, si  $n = 5$ ,  $S = \{1, 2, 5\}$  et  $\mathbf{y}$  est l'affectation partielle  $y_1 = 1, y_2 = 0$  et  $y_5 = 1$  alors l'affectation de  $\mathbf{x} = (1, 1, 1, 0, 0)$  par  $\mathbf{y}$  va être le vecteur  $\mathbf{z} = (1, 0, 1, 0, 1)$ .

Soit une fonction pseudo-booléenne  $f$ , un vecteur  $\mathbf{x} \in \mathbf{B}^n$  et une affectation partielle  $\mathbf{y} \in \mathbf{B}^n$  pour un sous-ensemble  $S \subseteq \mathbf{V}$ , alors on a les définitions suivantes.

**Persistance forte** Pour  $f$  aux valeurs de  $\mathbf{y}$ , si pour tout  $\mathbf{x} \in \text{Argmin}_{\mathbf{B}^n}(f)$ , on a  $\mathbf{x}[S] = \mathbf{y}$ . En d'autres termes, on dit qu'une affectation est persistante forte si

Optimisation de formes en sciences de l'ingénieur

l'ensemble des valeurs de l'affectation  $\mathbf{y}$  appartient à tous les minima globaux de la fonction  $f$ .

**Persistence faible** Pour  $f$  aux valeurs de  $\mathbf{y}$ , si pour tout  $\mathbf{x} \in \text{Argmin}_{\mathbf{B}^n}(f)$ , on a  $\mathbf{x}[S \leftarrow \mathbf{y}] \in \text{Argmin}_{\mathbf{B}^n}(f)$ . On dit qu'une affectation est persistante faible si toutes les valeurs de  $\mathbf{x}$  appartenant à tous les minima globaux de la fonction  $f$  le sont encore après affectation.

**Exemple 5.1.** Pour illustrer les deux notions précédentes, reprenons la fonction  $f_1$  de l'exemple 5.1 et sa posiforme  $\phi_{11}$ . On considère l'affectation partielle  $\mathbf{y}^* = (0) \in \mathbf{B}^{\{1\}}$ . On peut constater avec la table 5.1 que pour tous les minima globaux,  $x_1 = (0)$ . L'affectation  $\mathbf{y}$  est donc persistante forte.

Soit  $\mathbf{z}^* = (0, 1) \in \mathbf{B}^{\{1,2\}}$  une autre affectation partielle. Les solutions  $\mathbf{x} = (0, 0, 0, 0)$  et  $\mathbf{x} = (0, 0, 0, 1)$  appartiennent au minimum global de la fonction  $f_1$ , pourtant leur deuxième valeur  $x_2$  est différente de 1. L'affectation n'est donc pas persistante forte. Néanmoins, l'application de l'affectation  $\mathbf{z}^*$  sur  $\mathbf{x} = (0, 0, 0, 0)$  et de  $\mathbf{x} = (0, 0, 0, 1)$  donne  $\mathbf{x} = (0, 1, 0, 0)$  et  $\mathbf{x} = (0, 1, 0, 1)$ . Ces deux vecteurs appartiennent bien au minimum global de la fonction donc cette affectation est persistante faible.

## 5.2.4 Fonction pseudo-booléenne quadratique

L'optimisation d'une fonction pseudo-booléenne quadratique est un problème abondamment étudié. L'une des principales propriétés de ce type d'optimisation est qu'il est possible de trouver en un temps polynomial une borne inférieure aux valeurs de la fonction appelée la *roof duality*. Une fois cette borne inférieure atteinte, il est possible d'en déduire une affectation partielle ayant comme propriété d'être persistante forte.

On nomme  $\mathcal{F}_2$  la famille des fonctions pseudo-booléennes quadratiques. Une fonction pseudo-booléenne quadratique peut être représentée par sa forme multilinéaire polynomiale unique :

$$f(\mathbf{x}) = c_0 + \sum_{i=1}^n c_i x_i + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j \quad (5.7)$$

ou par une posiforme quadratique de la forme suivante :

$$\phi(\mathbf{x}) = a_0 + \sum_{u \in \mathbf{X}} a_u u + \sum_{u, v \in \mathbf{X}, u \neq v} a_{uv} uv \quad (5.8)$$

où  $a_u \geq 0$  et  $a_{uv} \geq 0$ . Généralement,  $a_0$  est nommé *terme constant*, les termes de degré 1 *terme linéaire*, et ceux de degré 2 *terme quadratique*. Comme dans (BOROS et Peter L HAMMER, 2002), le terme constant  $a_0$  de la posiforme  $\phi$  est noté  $C(\phi)$ . On notera aussi  $\mathcal{P}_2(f)$  la famille des posiformes quadratiques représentant la même fonction  $f$ .

### 5.2.5 Roof duality

Trouver le minimum d'une fonction pseudo-booléenne quadratique quelconque est un problème *NP-complet*, mais on peut calculer une borne inférieure aux valeurs de la fonction appelée *roof duality* en un temps polynomial. Une façon intuitive d'interpréter la *roof duality* est de trouver quelle est la posiforme  $\phi$  dans un ensemble de fonctions pseudo-booléennes vérifiant la même table de vérité, dont le terme constant  $C(\phi)$  est le plus grand. En effet, les coefficients d'une posiforme étant tous positifs, le terme constant maximum de  $\phi$  est une borne inférieure de  $f$ .

Ce problème a été étudié en optimisation de fonctions pseudo-booléennes quadratiques et il existe plusieurs techniques permettant de calculer cette borne inférieure de la fonction. On la note :

- $M_2(f)$  quand elle est obtenue par majorisation,
- $C_2(f)$  par la complémentation,
- $L_2(f)$  par la linéarisation,
- autres quand on utilise le Lagrangien, *paved duality*, etc.

Un résultat important est que toutes ces bornes sont les mêmes et égales à  $\max_{\phi \in \mathcal{P}_2(f)} C(\phi)$ . Les méthodes pour les obtenir sont donc équivalentes.

**Théorème 5.1.** Pour toutes fonctions pseudo-booléennes quadratiques  $f \in \mathcal{F}_2$ , on a (voir (BOROS et Peter L HAMMER, 2002)) :

$$M_2(f) = C_2(f) = L_2(f) \leq \min_{\mathbf{x} \in \mathbf{B}^n} f(\mathbf{x}).$$

Une fois la *roof duality* atteinte (c'est à dire après avoir transformé la posiforme d'origine en une posiforme avec la plus grande constante possible) la propriété suivante permet d'extraire un sous-ensemble persistant fort.

**Théorème 5.2. (Persistence forte)** Soit une fonction pseudo-booléenne quadratique  $f \in \mathcal{F}_2$ , soit  $\phi \in \mathcal{P}_2(f)$  une posiforme la représentant tel que  $C(\phi) = C_2(\phi)$ , si  $a_u > 0$  pour des labels  $u \in \mathbf{X}$ , alors  $u = 0$  dans tous les vecteurs binaires  $\mathbf{x} \in \text{Argmin}(f)$  minimisant  $f$  (voir (BOROS et Peter L HAMMER, 2002)).

Autrement dit, s'il reste un terme linéaire dans la formule obtenue après avoir atteint la *roof duality*, alors chaque affectation annulant le terme linéaire appartiendra à tous les minima globaux.

### 5.2.6 Réduction

Trouver la *roof duality* est un problème de classe polynomiale. En effet, elle peut être obtenue en résolvant un problème de programmation linéaire. Comme précédemment indiqué, plusieurs méthodes existent pour calculer cette borne. Dans cette partie, nous allons détailler une méthode fondée sur la théorie des graphes. Cette méthode est très efficace car fondée sur des algorithmes de la théorie des graphes très étudiés et optimisés. Ces algorithmes sont donc parmi les plus rapides pour les problèmes discrets.

L'algorithme est décomposé en plusieurs étapes : tout d'abord, la posiforme est traduite sous la forme d'un graphe dit induit. Ensuite, il faut effectuer la réduction de la fonction pseudo-booléenne pour obtenir la valeur de la *roof duality*. Cela se fait en pratique par poussage de flot sur le graphe induit et conduit à la détermination d'un graphe résiduel. Enfin, les affectations persistantes fortes sont obtenues à partir du parcours du graphe réduit.

### 5.2.6.1 Construction du graphe induit

Soit une fonction pseudo-booléenne quadratique  $f \in \mathcal{F}_2$  donnée par la posiforme  $\phi \in P_2(f)$  de forme (5.8). On associe à cette posiforme quadratique un graphe orienté  $G_\phi = (N, A)$  où l'ensemble de sommets est défini par  $N = \mathbf{X} \cup \{0, \bar{0}\}$ .

Il y a deux sommets par variable : le sommet  $x_i$ , noté  $i$  dans le graphe, représentant le cas  $x_i = 1$  et le sommet  $\bar{x}_i$ , noté  $\bar{i}$  dans le graphe, représentant le cas  $\bar{x}_i = 1$ , plus deux sommets  $x_0$  et  $\bar{x}_0$  représentant les cas  $x_0 = 1$  et  $\bar{x}_0 = 1$ . Ces deux sommets sont notés  $0$  et  $\bar{0}$ , respectivement, dans le graphe induit.

Les arcs sont définis à partir des termes de la posiforme. À chaque terme quadratique  $a_{uv}uv$  correspond deux arcs  $(\overrightarrow{u, \bar{v}})$  et  $(\overrightarrow{\bar{v}, u})$  avec chacun un poids  $\frac{1}{2}a_{uv}$ . À chaque terme linéaire  $a_u u$  correspond aussi deux arcs  $(\overrightarrow{u, \bar{x}_0})$  et  $(\overrightarrow{x_0, \bar{u}})$  avec chacun comme poids  $\frac{1}{2}a_u$  (voir (BOROS et Peter L HAMMER, 2002)).

**Exemple 5.2.** Soit la posiforme quadratique suivante :

$$\phi_2(\mathbf{x}) = 10x_1 + 8\bar{x}_1x_2 + 6\bar{x}_2x_3 + 4\bar{x}_3. \quad (5.9)$$

Le premier terme va induire deux arcs orientés  $(\overrightarrow{x_1, \bar{x}_0})$  et  $(\overrightarrow{x_0, \bar{x}_1})$  de poids 5.

Le deuxième terme va induire deux arcs orientés  $(\overrightarrow{\bar{x}_1, \bar{x}_2})$  et  $(\overrightarrow{x_2, x_1})$  de poids 4.

Le troisième terme va induire deux arcs orientés  $(\overrightarrow{\bar{x}_2, \bar{x}_3})$  et  $(\overrightarrow{x_3, x_2})$  de poids 3.

Le quatrième terme va induire deux arcs orientés  $(\overrightarrow{\bar{x}_3, \bar{x}_0})$  et  $(\overrightarrow{x_0, x_3})$  de poids 2.

Le graphe induit est montré dans la figure 5.1(a).

### 5.2.6.2 Calcul de la borne inférieure par flot maximum

Une fois le graphe induit construit, nous allons voir qu'il y a équivalence entre une somme alternée dans une équation pseudo-booléenne et un chemin augmentant le flot dans le graphe induit de la posiforme  $\phi$ . La caractéristique d'une somme alternée est qu'il est possible de faire apparaître, par une identité algébrique, une constante, et par conséquent, d'augmenter le terme constant  $C(\phi)$  de la posiforme pour atteindre la valeur de *roof duality*  $C_2(\phi)$ . Cette réduction est réalisée en pratique par un algorithme de poussage de flot sur le graphe induit grâce à l'équivalence entre somme alternée et chemin augmentant.



### 5.2.6.3 Équivalence entre somme alternée et chemin augmentant

Un flot possible dans un graphe  $G = (N, A)$  avec comme source  $x_0$  et comme puits  $\bar{x}_0$  est une application  $\zeta : A \rightarrow \mathbb{R}_+$  respectant les contraintes :

$$\zeta(u, v) \leq c_{u,v} \quad \text{pour tous les arcs } (\overrightarrow{u, v}) \in A, \text{ et} \quad (5.10)$$

$$\sum_{(\overrightarrow{u, v}) \in A} \zeta(u, v) = \sum_{(\overrightarrow{v, w}) \in A} \zeta(v, w) \quad \text{pour tous les sommets } v \in \mathbf{X}. \quad (5.11)$$

La première contrainte indique que le flot ne peut pas dépasser la capacité de chaque arc. La seconde contrainte impose que le flot total entrant soit égal au flot total sortant en chaque sommet.

Pour un graphe donné  $G = (N, A)$  et un flot possible  $\zeta$  dans ce graphe, on appelle graphe réduit  $G[\zeta] = (N, A^\zeta)$  avec comme capacités :

$$c_{uv}^\zeta = \begin{cases} c_{uv} - \zeta(u, v) & \text{pour } (\overrightarrow{u, v}) \in A \\ \zeta(u, v) & \text{pour } (\overrightarrow{v, u}) \in A \end{cases} \quad (5.12)$$

Quand  $u_1, u_2, \dots, u_k \in \mathbf{X}$ , on appelle somme alternée, la forme quadratique suivante :

$$u_1 + \bar{u}_1 u_2 + \bar{u}_2 u_3 + \dots + \bar{u}_{k-1} u_k + \bar{u}_k. \quad (5.13)$$

On remarque que cette forme quadratique est bien homogène et alternée en introduisant  $\bar{x}_0$  et  $x_0$ . Une posiforme quadratique  $\phi$  contient une somme alternée (5.13) de poids  $\omega$  si nous avons  $a_{u_1} \geq \omega, a_{\bar{u}_j u_{j+1}} \geq \omega$  pour  $j = 1, \dots, k-1$  et  $a_{\bar{u}_k} \geq \omega$  pour tous les coefficients correspondants de  $\phi$  (voir (BOROS et Peter L HAMMER, 2002)).

**Proposition 5.1.** L'identité suivante est vérifiée pour les sommes alternées :

$$u_1 + \bar{u}_1 u_2 + \dots + \bar{u}_{k-1} u_k + \bar{u}_k = 1 + u_1 \bar{u}_2 + \dots + u_{k-1} \bar{u}_k. \quad (5.14)$$

Cela se démontre en changeant  $x_i$  en  $1 - \bar{x}_i$ . Avec cette identité, une posiforme quadratique  $\phi$  qui contient une somme alternée (5.13) de poids  $\omega$  peut être transformée en une posiforme équivalente ayant un terme constant plus important. Pour ce faire, il faut dans un premier temps réécrire  $\phi$  en :

$$\phi = \omega[u_1 + \bar{u}_1 u_2 + \bar{u}_2 u_3 + \dots + \bar{u}_{k-1} u_k + \bar{u}_k] + \phi'.$$

Par construction,  $\phi'$  est aussi une posiforme quadratique. Par conséquent, en appliquant l'identité (5.14), on obtient :

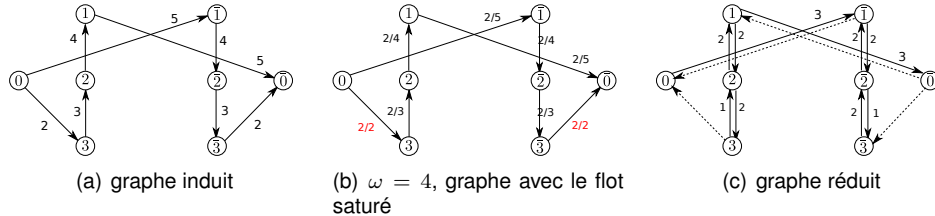
$$\phi = \omega + \omega[u_1 + \bar{u}_1 u_2 + \bar{u}_2 u_3 + \dots + \bar{u}_{k-1} u_k + \bar{u}_k] + \phi'.$$

En construisant le graphe induit d'une somme alternée suivant les règles précédentes, on voit qu'il y a une correspondance entre une somme alternée contenue dans une posiforme  $\phi$  de poids  $\omega$  et un chemin augmentant de capacité  $\omega$  dans le graphe induit de  $\phi$ . La proposition suivante peut donc être énoncée :

Optimisation de formes en sciences de l'ingénieur

**Figure 5.1**

Équivalence entre la réduction de la posiforme et le poussage de flot sur le graphe induit. (a) montre le graphe induit par la posiforme de l'exemple 5.2. (b) montre le poussage de flot maximum. (c) montre le graphe réduit.



**Proposition 5.2.** On considère une posiforme  $\phi \in P_2(f)$  et un flot possible  $\zeta$  dans le graphe  $G = G_\phi$ . Alors  $x_0, u_1, \dots, u_k, \bar{x}_0$  est un chemin augmentant de capacité  $\omega > 0$  dans ce flot si et seulement si  $u_1 + \bar{u}_1 u_2 + \dots + \bar{u}_{k-1} u_k + \bar{u}_k$  est une somme alternée de poids  $\omega$  dans la posiforme  $\phi$ .

**Exemple 5.2.** Reprenons la posiforme  $\phi_2$  de l'exemple précédent. Son graphe induit est montré en figure 5.1(a). Nous allons illustrer l'équivalence entre la capacité d'un chemin augmentant le flot sur le graphe induit et le poids d'une somme alternée utilisée lors de la réduction de la posiforme.

Tous les coefficients de  $\phi_2$  sont supérieurs ou égaux à 4. On choisit donc  $\omega = 4$  et on effectue le groupement suivant :

$$\phi_2(\mathbf{x}) = \underbrace{4}_{\omega} \underbrace{(x_1 + \bar{x}_1 x_2 + \bar{x}_2 x_3 + \bar{x}_3)}_{\text{somme alternée}} + \underbrace{6x_1 + 4\bar{x}_1 x_2 + 2\bar{x}_2 x_3}_{\text{forme résiduelle } \phi'} \quad (5.15)$$

$$= 4(1 + x_1 \bar{x}_2 + x_2 \bar{x}_3) + 6x_1 + 4\bar{x}_1 x_2 + 2\bar{x}_2 x_3 \quad (5.16)$$

$$= 4 + \underbrace{6x_1 + 4x_1 \bar{x}_2 + 4\bar{x}_1 x_2 + 2\bar{x}_2 x_3 + 4x_2 \bar{x}_3}_{\text{forme réduite } \psi} \quad (5.17)$$

En appliquant l'identité (5.14) sur la somme alternée qui est maintenant dans (5.15), un terme constant apparaît dans l'équation (5.16). Ceci a comme conséquence, une fois (5.16) développée, d'atteindre la valeur de *roof duality* dans ce cas. Ces manipulations algébriques ont leur équivalent en terme de poussage de flot sur le graphe. En effet, à partir de la figure 5.1(b), on voit que le flot maximum jusqu'à saturation est obtenu en faisant passer un flot de 2 dans les deux chemins possibles entre 0 et 0-bar. Ces deux chemins qui sont symétriques correspondent à la somme alternée dans l'équation (5.15). On peut aussi constater que  $\omega = 2 + 2 = 4$ . En construisant le graphe réduit avec (5.12) et en simplifiant au niveau des arcs qui partent ou arrivent de  $x_0$  et  $\bar{x}_0$  (cela correspond à la réduction des termes linéaires par complémentation), on arrive au graphe réduit affiché en figure 5.1(c). On voit qu'il correspond exactement à la forme réduite  $\psi$  dans l'équation (5.17). Les arcs en pointillés ont une capacité nulle.

### 5.2.6.4 Lien entre le flot maximum et la valeur de *roof duality*

La réduction précédente doit être réalisée jusqu'à ce qu'il ne soit plus possible de factoriser de nouvelles sommes alternées. Alors, la propriété fondamentale de cette réduction est que la valeur *roof duality*  $C_2(f)$  est atteinte en sommant le terme constant de la posiforme initiale avec la valeur du flot maximum atteinte par l'algorithme de poussage de flot.

**Théorème 5.3.** Soit une fonction pseudo-booléenne quadratique  $f$ , et une posiforme quadratique la représentant  $\phi \in P_2(f)$ , et  $\zeta^*$  la valeur du flot maximum dans  $G_\phi$ , alors :

$$C_2(f) = C(\phi) + \zeta^*.$$

Après avoir calculé le flot maximum, et donc la valeur de *roof duality*, il est possible de trouver les affectations persistantes fortes. Une façon efficace est d'utiliser le graphe réduit.

**Théorème 5.4 (Persistance forte et flot maximum).** Soit  $\phi \in P_2(f)$  pour une fonction pseudo-booléenne quadratique  $f$ , soit  $\zeta^*$  un flot maximum dans  $G = G_\phi$ , soit  $S \in L$  l'ensemble des sommets dans  $G$  qui peuvent être atteints en partant de  $x_0$  par un chemin avec une capacité résiduelle positive, alors  $u(\mathbf{x}^*) = 1$  pour tout  $u \in S$  et pour tous les vecteurs  $\mathbf{x}^* \in \text{Argmin}(f)$ .

En d'autres termes, pour toutes variables pouvant être atteintes dans le graphe réduit en partant de la source  $x_0$  par des capacités strictement positives, leur affectation à 1 est persistante forte.

### 5.2.6.5 Récapitulatif

Pour résumer, les affectations persistantes fortes pour une fonction pseudo-booléenne quadratique  $f$  sont calculées de la façon suivante :

- Trouver une posiforme quadratique  $\phi$  de même table de vérité que  $f$ .
- Construire le graphe induit  $G_\phi$  de  $\phi$ .
- Calculer le flot maximum  $\zeta^*$  de  $G_\phi$ .
- Construire le graphe réduit  $G_\psi$  de  $G_\phi$  à partir du flot maximum  $\zeta^*$ .
- Pour tous les sommets pouvant être atteints à partir de la source  $x_0$ , affecter les variables correspondantes à 1.

**Exemple 5.3.** La figure 5.2(a) montre le graphe induit. En observant la figure 5.2(a), on peut constater qu'il est possible de faire passer 6 flots de poids 1 sur chacun des chemins suivants :

$$x_0 \rightarrow x_1 \rightarrow \bar{x}_2 \rightarrow \bar{x}_3 \rightarrow \bar{x}_0 \text{ et } x_0 \rightarrow x_3 \rightarrow x_2 \rightarrow \bar{x}_1 \rightarrow \bar{x}_0, \quad (5.18)$$

$$x_0 \rightarrow x_3 \rightarrow \bar{x}_4 \rightarrow \bar{x}_0 \text{ et } x_0 \rightarrow x_4 \rightarrow \bar{x}_3 \rightarrow \bar{x}_0, \quad (5.19)$$

$$x_0 \rightarrow x_3 \rightarrow \bar{x}_4 \rightarrow \bar{x}_5 \rightarrow \bar{x}_0 \text{ et } x_0 \rightarrow x_5 \rightarrow x_4 \rightarrow \bar{x}_3 \rightarrow \bar{x}_0. \quad (5.20)$$

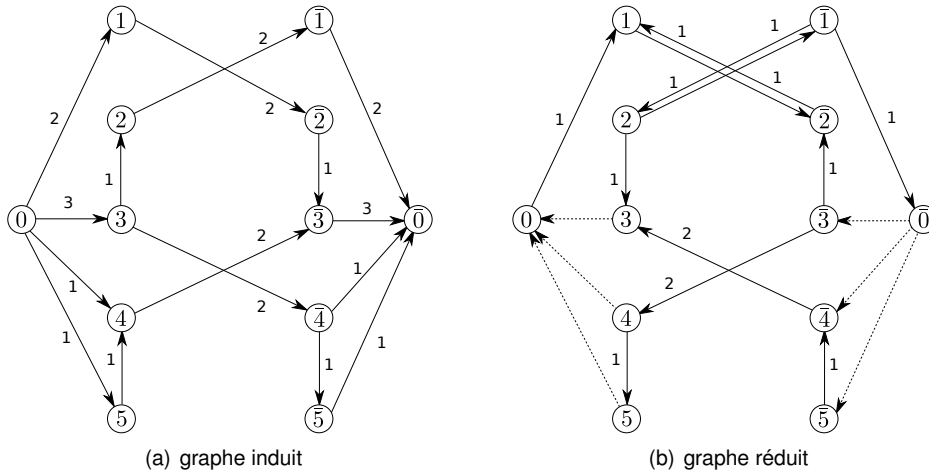
Soit la fonction pseudo-booléenne quadratique suivante :

$$f_3(\mathbf{x}) = 10 - 4x_1 - 4x_3 - 2x_4 + 4x_4x_2 - 2x_2x_3 + 4x_3x_4 - 2x_4x_5.$$

Optimisation de formes en sciences de l'ingénieur

**Figure 5.2**

(a) montre le graphe induit de l'exemple 5.3 et (b) le graphe réduit.



En substituant avec l'identité  $x = 1 - \bar{x}$  chaque terme quadratique ayant un coefficient négatif, on a la posiforme suivante :

$$\phi_3(\mathbf{x}) = -4 + 4\bar{x}_1 + 6\bar{x}_3 + 2\bar{x}_4 + 2\bar{x}_5 + 4x_1x_2 + 2\bar{x}_2x_3 + 4x_3x_4 + 2\bar{x}_4x_5.$$

Ces flots pouvant être poussés séquentiellement, ce sont des chemins augmentants. Ces chemins augmentants correspondent respectivement aux sommes alternées suivantes :

$$\bar{x}_1 + x_1x_2 + \bar{x}_2x_3 + \bar{x}_3 \text{ et } \bar{x}_3 + x_3\bar{x}_2 + x_2x_1 + \bar{x}_1, \quad (5.21)$$

$$\bar{x}_3 + x_3x_4 + \bar{x}_4 \text{ et } \bar{x}_4 + x_4x_3 + \bar{x}_3, \quad (5.22)$$

$$\bar{x}_3 + x_3x_4 + \bar{x}_4x_5 + \bar{x}_5 \text{ et } \bar{x}_5 + x_5\bar{x}_4 + x_4x_3 + \bar{x}_3. \quad (5.23)$$

On remarque qu'il n'y a pas d'autre chemin augmentant dans le graphe Fig. 5.2(a). Le nombre de chemins augmentants maximum a été atteint, et la valeur du flot maximum est donc de  $\zeta^* = 6$ . Le graphe réduit peut être construit comme montré en figure 5.2(b). Le graphe réduit correspond à la posiforme quadratique suivante :

$$\psi_3(\mathbf{x}) = 2\bar{x}_1 + 2x_1x_2 + 2\bar{x}_1\bar{x}_2 + 2x_2\bar{x}_3 + 4\bar{x}_3\bar{x}_4 + 2x_4\bar{x}_5$$

qu'il est aussi possible de retrouver en faisant algébriquement la réduction sur la posiforme  $\phi_3$ .

Comme le flot maximum est  $\zeta^* = 6$ , on a :  $\phi_3(\mathbf{x}) = C(\phi_3) + \zeta^* + \psi_3(\mathbf{x}) = 2 + \psi_3(\mathbf{x})$ , d'où  $C_2(f_3) = 2$ . Le graphe réduit figure 5.2(b) a deux sommets directement atteignables à partir de la source : le sommet 1 et  $\bar{2}$ . D'après le théorème 5.4, les affectations  $x_1 = 1$  et  $x_2 = 0$  sont donc vérifiées pour chaque minimum global de  $f_3$ . On peut donc affecter ces valeurs à  $\psi_3(\mathbf{x})$ . Ainsi, minimiser  $f_3$  équivaut maintenant à minimiser  $\psi_3^*(\mathbf{x}) = 4\bar{x}_3\bar{x}_4 + 2x_4\bar{x}_5$ . Les affectations  $(\mathbf{x})$  qui annulent cette fonction sont persistantes faibles. Ceci se vérifie sur la table 5.2.

**TABLE 5.2**

Table de vérité de la fonction pseudo-booléenne de l'exemple 5.3. Les minimums de  $f_3$  sont en gras.

<b>x</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
$f_3(\mathbf{x})$	10	10	8	6	6	6	8	6	10	10	8	6	4	4	6	4
<b>x</b>	1	1	1	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	1	1	1	1	1	1	1	
	0	0	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	1	1	1	1	1	1	1	
	0	0	0	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	0	0	0	0	1	1	1	
	0	0	1	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	0	0	1	1	0	0	1	
	0	1	0	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	0	1	0	1	0	1	0	
$f_3(\mathbf{x})$	6	6	4	<b>2</b>	<b>2</b>	<b>2</b>	4	<b>2</b>	10	10	8	6	4	4	6	4

## 5.2.7 Cas sous-modulaire

### 5.2.7.1 Définition

Une fonction  $f$  sur les ensembles est sous-modulaire si :

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \quad \forall X, Y \quad (5.24)$$

est respecté pour tous les sous-ensembles  $X, Y$ . Le pendant algébrique de cette définition, pour une fonction pseudo-booléenne  $f : \mathbf{B}^n \rightarrow \mathbb{R}$  quadratique, est :

$$\frac{\partial f}{\partial x_i \partial x_j}(\mathbf{x}) \leq 0. \quad (5.25)$$

Autrement dit, une fonction pseudo-booléenne quadratique est sous-modulaire si, sous sa forme multilinéaire polynomiale unique (5.7), ses dérivées secondes sont toutes négatives ou nulles. Cela implique que tous ses termes quadratiques ont un coefficient de signe négatif. Nous allons maintenant voir que trouver le minimum global d'une fonction sous-modulaire est aussi équivalent à la recherche de la coupe minimale d'un graphe, avec certaines simplifications par rapport au cas général de la *roof duality* vu précédemment.

Optimisation de formes en sciences de l'ingénieur

### 5.2.7.2 Posiforme quadratique sous-modulaire

Quand une fonction est sous-modulaire, il est toujours possible de l'écrire comme une posiforme  $\phi$  sous la forme (à une constante près) :

$$\phi(\mathbf{x}) = \sum_{i \in P} a_i x_i + \sum_{j \in N} a_j \bar{x}_j + \sum_{1 \leq i < j \leq n} a_{ij} x_i \bar{x}_j \quad (5.26)$$

où  $P, N \subseteq \mathbf{V}$  et tous les coefficients  $a_i$  ( $i \in P \cup N$ ) et  $a_{ij}$  ( $1 \leq i < j \leq n$ ) sont positifs.

### 5.2.7.3 Construction du graphe induit

Comme expliqué dans la partie 5.2.6.1 précédente, il est possible de construire le graphe induit de la posiforme 5.26. Comme il n'y a pas de terme en  $x_i x_j$  ou  $\bar{x}_i \bar{x}_j$ , il n'y a pas d'arcs entre les sommets  $i$  et  $\bar{i}$  dans le graphe induit et par conséquent, le graphe réduit peut être partagé en deux graphes indépendants et symétriques l'un de l'autre. Cela implique qu'il est possible d'associer à  $\phi$  le graphe plus simple  $N_\phi = (V, A)$  construit de la façon suivante :

- l'ensemble des sommets est  $V = \{s, t\} \cup \mathbf{V}$ .
- l'ensemble des arcs est

$$A = \{ \overrightarrow{(s, j)} \mid j \in N \} \cup \{ \overrightarrow{(i, t)} \mid i \in P \} \cup \{ \overrightarrow{(i, j)} \mid 1 \leq i < j \leq n \}$$

où les capacités des arcs sont respectivement  $c_{sj} = a_j$  pour  $j \in N$ ,  $c_{it} = a_i$  pour  $i \in P$  et  $c_{ij} = a_{ij}$  pour  $1 \leq i < j \leq n$ .

En d'autres termes, l'ensemble des sommets  $V$  est composé de l'ensemble des indices  $\mathbf{V}$  ainsi qu'une source  $s$  et un puits  $t$ . On construit, pour chaque terme unitaire, un arc. Quand le terme est une variable  $x_i$ , un arc du sommet  $i$  au puits  $t$  est créé, avec comme poids le coefficient  $a_i$ . Quand le terme est le complément d'une variable  $\bar{x}_j$ , un arc de la source  $s$  au sommet  $j$  est créé, avec comme poids le coefficient  $a_j$ . Enfin, on construit pour chaque terme  $a_{ij} x_i \bar{x}_j$  un arc entre  $i$  et  $j$ . C'est ce type de graphe induit qui est construit plus habituellement dans les méthodes dites par coupe de graphe, voir par exemple (KOLMOGOROV et ZABIH, 2002).

### 5.2.7.4 Minimisation par coupe de graphe

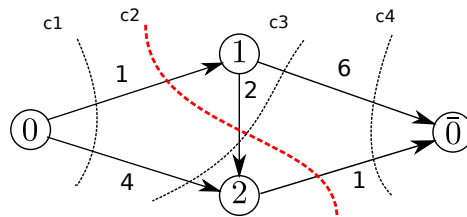
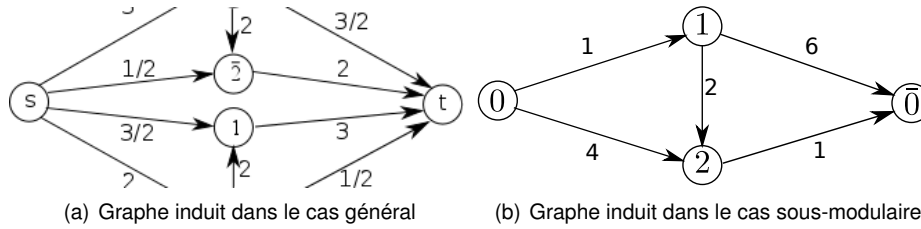
Il y a une correspondance entre une coupe qui sépare la source  $s$  et le puits  $t$  dans le graphe et le vecteur binaire  $\mathbf{x} \in \mathbf{B}^n \leftrightarrow S_{\mathbf{x}} \stackrel{\text{def}}{=} \{s\} \cup \{j \mid x_j = 1\}$ . Il est donc facile de voir qu'avec cette définition, on a pour tout  $\mathbf{x} \in \mathbf{B}^n$  :

$$\phi(\mathbf{x}) = \sum_{\substack{i \in S_{\mathbf{x}} \\ j \notin S_{\mathbf{x}}}} a_{i,j} \quad (5.27)$$

avec la convention que seuls les arcs traversants de  $s$  à  $t$  comptent. Un minimum de  $\phi(\mathbf{x})$  correspond à une coupe de poids minimal de  $N_\phi$ . Une coupe minimal

**Figure 5.3**

Graphes induit dans le cas sous-modulaire quadratique de l'exemple 5.4.



(c) Coupes possibles de poids  $w(c_1) = 7$ ,  $w(c_2) = 6$ ,  $w(c_3) = 10$  et  $w(c_4) = 7$ .

**TABLE 5.3**

Table de vérité de la fonction pseudo-booléenne de l'exemple 5.4. Le minimum est en gras.

	0	<b>0</b>	1	1
	0	<b>1</b>	0	1
$f_4(\mathbf{x})$	7	<b>6</b>	10	7

peut être trouvée par une méthode de maximisation du flot à travers le graphe  $N_\phi$ . Cela permet donc de trouver une affectation de toutes les variables de  $\mathbf{x}$  qui atteint la valeur minimale de  $\phi$ .

**Exemple 5.4.** On cherche à minimiser la posiforme suivante :

$$\phi_4(\mathbf{x}) = 3\bar{x}_1 + 6x_1 + 4\bar{x}_2 + x_2 + 2x_2\bar{x}_1. \tag{5.28}$$

Il est aisé de vérifier que cette posiforme est sous-modulaire.

En se plaçant dans le cas général, avec les règles de la partie 5.2.6.1, on peut construire le graphe induit montré en figure 5.3(a). Ce graphe se décompose clairement en deux graphes indépendants. En suivant les règles précédentes, on construit alternativement le graphe induit simplifié dans le cas sous-modulaire, comme montré en figure 5.3(b).

La figure 5.3(c) montre les quatre coupes possibles. Elles sont respectivement de poids 7, 6, 10 et 7. Par exemple pour la coupe  $c_2$  de la figure 5.3(c), son poids

Optimisation de formes en sciences de l'ingénieur

est de  $3 + 2 + 1 = 6$  car les arcs  $(s, 1)$ ,  $(1, 2)$   $(2, t)$  sont coupés, tous en allant de la source au puits. Pour la coupe  $c_3$ , son poids est  $4 + 6 = 10$  car seuls les arcs  $(s, 2)$  et  $(1, t)$  sont coupés en allant de la source au puits. Les capacités des 4 coupes correspondent bien aux valeurs de la table de vérité de  $\phi_4$  en table 5.3. En conséquence, la capacité de la coupe minimale dans le graphe induit figure 5.3(b) est égale à la valeur minimum de la fonction  $\phi_4$  dans (5.28).

La coupe minimal peut être obtenue en cherchant le flot maximum. Dans cet exemple, on voit que l'on peut faire passer un flot de valeur 2 de  $s$  à  $t$  en passant par les sommets 2 puis 1, un flot de 1 en passant par le sommet 2 et un flot de 3 en passant par le sommet 1. On vérifie donc que les 3 arcs coupés par la coupe  $c_2$  sont bien saturés par le flot maximum.

Avec la coupe minimum, seule le sommet 2 est du côté de la source  $s$ , donc  $x_2 = 1$ . Inversement, le sommet 1 étant du côté du puits  $t$ , on a  $x_1 = 0$ . On retrouve bien la solution qui atteint la valeur minimum de  $\phi_4$  dans la table 5.3. On peut vérifier que l'on trouve le même résultat en procédant sur le graphe induit dans le cas général.

### 5.2.8 Optimisation de fonctions pseudo-booléennes de degré supérieur ou égal à 3

Il est difficile d'optimiser une fonction pseudo-booléenne de degré supérieur ou égal à 3, à cause de la présence de minima locaux, surtout, dans le cas non sous-modulaire. Pour optimiser de telles fonctions, deux approches ont néanmoins été proposées. La première approche consiste à décomposer une fonction de degré quelconque en fonction pseudo-booléenne quadratique possédant le même minimum et à l'optimiser avec la méthode par *roof duality*. La seconde approche est de généraliser directement le principe de *roof duality* au cas de degré strictement supérieur à deux (KAHL et STRANDMARK, 2011). Ceci est difficile. Nous allons maintenant seulement résumer la méthode proposée dans (ISHIKAWA, 2009) qui est fondée sur le principe de substitution.

#### 5.2.8.1 Substitution

Une première méthode pour décomposer une fonction pseudo-booléenne d'ordre  $n$  à l'ordre 2 à été introduite en 1975 (ROSENBERG, 1975). Elle consiste à ajouter une variable auxiliaire afin de décomposer les termes de grands ordres en plusieurs termes d'ordres inférieurs. Cette méthode est itérative et fonctionne pour un degré quelconque.

Cette méthode de substitution qui permet, en remplaçant le produit de deux variables par une variable auxiliaire, de réduire le degré maximum d'une fonction pseudo-booléenne. On considère ces deux équivalences :

$$xy = z \iff xy - 2xz - 2yz + 3z = 0 \tag{5.29}$$

$$xy \neq z \iff xy - 2xz - 2yz + 3z > 0 \tag{5.30}$$



**TABLE 5.4**

Tables de vérité des fonctions pseudo-booléennes de l'exemple 5.5. Le minimum est en gras.

<b>x</b>	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
$f_5(\mathbf{x})$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>-1</b>

<b>x</b>	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
$f'_5(\mathbf{x})$	0	<b>18</b>	0	<b>15</b>	0	<b>6</b>	0	<b>3</b>	0	<b>6</b>	0	<b>3</b>	<b>8</b>	<b>2</b>	<b>8</b>	<b>2</b>	<b>8</b>	<b>2</b>	<b>-1</b>

Les deux équivalences peuvent être facilement vérifiées en faisant les tables de vérité.

Il est donc possible, dans une fonction pseudo-booléenne, de remplacer le produit  $xy$  par  $z$  dans un terme de degré supérieur à deux, et d'ajouter l'expression  $xy - 2xz - 2yz + 3z$  comme un terme de pénalité, avec un grand coefficient multiplicatif. Alors, d'après (ROSENBERG, 1975), le vecteur minimisant la nouvelle équation permet d'obtenir le vecteur minimisant l'équation de départ en omettant les variables introduites en plus, ce qui se traduit par :

$$\min_{w,x,y} wxy = \min_{w,x,y,z} wz + \mu(xy - 2xz - 2yz + 3z) \quad (5.31)$$

où  $w, x, y$  sont les variables de départ et  $z$  est la variable ajoutée.  $\mu$  est le coefficient multiplicatif. Il doit être assez grand, pour cela, il peut être initialisé avec  $\mu = 1 + \sum_{S \in \mathbf{V}} |c_S|$ . Cette étape de substitution peut être itérée jusqu'à ce que le degré maximum du résultat soit égal à deux.

**Exemple 5.5.** Soit la fonction pseudo-booléenne :

$$f_5(x_1, x_2, x_3) = 2x_1x_2 - 3x_1x_2x_3. \quad (5.32)$$

En substituant  $x_1x_2$  par une nouvelle variable  $x_4$  avec  $\mu = 1 + 2 + 3 = 6$ , la fonction (5.32) se réécrit en :

$$\begin{aligned} f'_5(x_1, x_2, x_3, x_4) &= 2x_1x_2 - 3 \underbrace{x_1x_2}_{x_4} x_3 + 6(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4) \\ &= 18x_4 + 8x_1x_2 - 3x_3x_4 - 12x_1x_4 - 12x_2x_4. \end{aligned} \quad (5.33)$$

En observant la table de vérité 5.4 de la forme originale  $f_5$  (5.32) et de la forme quadratique  $f'_5$  (5.33), on constate que le minimum de  $f'_5$  est atteint pour le

Optimisation de formes en sciences de l'ingénieur

vecteur  $\mathbf{x} = (1111)$ . Si on omet la variable ajoutée  $x_4$  lors de la décomposition, le vecteur des variables d'origine est alors  $\mathbf{x} = (111)$  et il permet bien d'atteindre le minimum global de  $f_5$ , comme on le vérifie facilement dans la table de vérité.

### 5.2.8.2 Autres substitutions

Le principal défaut de la méthode par substitution précédente, est que, pour chaque substitution, un terme non sous-modulaire est ajouté. Au fil des itérations, le nombre de termes non sous-modulaires augmente donc. Or, le nombre d'affectations persistantes fortes lors du calcul de la *roof duality* dépend directement du nombre et des coefficients des termes quadratiques non sous-modulaires dans la forme multilinéaire polynomiale unique. Comme constaté dans (GALLAGHER, BATRA et PARIKH, 2011), plus il y a de termes non sous-modulaire, et plus leurs coefficients sont importants, et moins il y a d'affectations fortes lors du calcul de la *roof duality*.

Après ce constat, de nouvelles méthodes ont été proposées afin de minimiser l'amplitude et le nombre des termes non sous-modulaire, donc de coefficient positifs, lors de la décomposition d'un degré quelconque en termes de degré quadratique (ISHIKAWA, 2009). En effet, plusieurs types de décompositions sont donc possibles pour un même terme (KOLMOGOROV et ZABIH, 2002; ROSENBERG, 1975; ISHIKAWA, 2009). Ces méthodes ont permis d'augmenter considérablement le nombre de persistances fortes lors du calcul de la *roof duality*, ce qui a permis de beaucoup mieux optimiser les problèmes d'origine. Néanmoins, le choix de la bonne substitution n'est pas simple. Dans (GALLAGHER, BATRA et PARIKH, 2011), il est donc proposé, pour chaque terme, de choisir entre différents types de décompositions. Pour cela, une technique d'inférence est proposée pour permettre de choisir le type de décomposition en minimisant soit, le nombre d'arcs non sous-modulaires, soit l'amplitude des termes non sous-modulaires dans la fonction pseudo-booléenne finale. Cette méthode a l'avantage de tirer parti de chaque type de substitution et d'augmenter le nombre d'affectations persistantes fortes.

D'autres méthodes ont par la suite été proposées pour améliorer le taux d'affectation et la rapidité de la réduction. En effet, il est possible de décomposer directement une fonction de degré quelconque en un degré quadratique. Dans (ISHIKAWA, 2011), la méthode par substitution proposée fonctionne quel que soit le degré de la fonction. Une autre méthode est proposée dans (FIX et al., 2011) par coupe de graphe.

## 5.3 Extension au cas multi-labels

### 5.3.1 Optimum global

Une des propriétés les plus remarquables est la possibilité de trouver en temps polynomial une solution qui minimise globalement une fonction pseudo-

booléenne sous-modulaire. Cette propriété a pu être étendue au cas multi-labels lorsque la fonction ou l'énergie (5.1) contient au maximum des termes à deux variables et que la fonction réelle associée à ces termes est convexe (ISHIKAWA, 2003).

### 5.3.2 Stratégie de fusion

L'optimisation multi-labels d'une fonction discrète peut être approchée comme une succession de sous-problèmes binaires. En effet, si nous prenons deux vecteurs de labels, il est possible de trouver grâce aux méthodes d'optimisation binaires précédentes, quelle est la combinaison des composantes de ces deux vecteurs qui minimise la fonction. Ce procédé est nommé une *fusion* (LEMPITSKY et al., 2010).

La stratégie la plus courante est de considérer le vecteur  $\mathbf{L}^t$  des labels de la solution courante et un nouveau vecteur de labels  $\mathbf{L}^p$ . L'objectif de la fusion est de trouver la meilleure sélection des labels  $\mathbf{L}^{t+1}$ , c'est à dire celle pour laquelle l'énergie (5.1) est la plus faible. Pour chaque variable, un choix booléen doit être réalisé. Afin de combiner les vecteurs  $\mathbf{L}^t$  et  $\mathbf{L}^p$ , la combinaison  $\mathbf{L}^b(B)$  est définie par la fonction booléenne auxiliaire suivante :

$$\mathbf{L}(\mathbf{x}) = \bar{\mathbf{x}} \cdot \mathbf{L}^t + \mathbf{x} \cdot \mathbf{L}^p \quad (5.34)$$

où  $\mathbf{x}$  est un vecteur de variables booléennes et où le produit noté avec un point est terme à terme. Il y a donc une variable binaire  $x_i$  par site  $i$ . Quand  $x_i = 0$ , le label courant  $L_i^t$  est conservé pour le site  $i$ . Au contraire, quand  $x_i = 1$ , c'est le label proposé  $L_i^p$  qui est sélectionné.

Une fois que le meilleur  $\mathbf{x}^*$  est obtenu par optimisation quadratique pseudo-booléenne, la meilleure combinaison de labels devient la solution courante :

$$\mathbf{L}^{t+1} = \mathbf{L}(\mathbf{x}^*). \quad (5.35)$$

Pour trouver  $\mathbf{x}^*$ , on doit minimiser la fonction suivante :

$$\phi(\mathbf{x}) = f(\mathbf{L}(\mathbf{x})). \quad (5.36)$$

Soit en utilisant la définition de  $f$  à partir de (5.1) :

$$\begin{aligned} \phi(\mathbf{x}) &= \sum_{u \in C_1} \vartheta_u(L_u^t) \bar{x}_u + \vartheta_u(L_u^p) x_u \\ &+ \sum_{u,v \in C_2} \vartheta_{uv}(L_u^t, L_v^t) \bar{x}_u \bar{x}_v + \vartheta_{uv}(L_u^t, L_v^p) \bar{x}_u x_v \\ &\quad + \vartheta_{uv}(L_u^p, L_v^t) x_u \bar{x}_v + \vartheta_{uv}(L_u^p, L_v^p) x_u x_v \\ &+ \sum_{u,v,w \in C_3} \dots \end{aligned} \quad (5.37)$$

Si la fonction pseudo-booléenne (5.37) est sous-modulaire, alors toutes les variables de  $\mathbf{x}$  pourront être déterminées à cette étape. Si elle n'est pas sous-modulaire, seul un sous-ensemble de variable sera déterminé, dans un premier

temps, en utilisant la *roof duality*. Dans un deuxième temps, les affectations persistantes faibles seront calculées en minimisant l'équation réduite. En cas d'indétermination, plusieurs stratégies sont possibles, la plus simple est de garder la solution courante.

Si la fonction (5.37) a un degré supérieur à 2, elle peut être décomposée en une forme de degré quadratique, et ensuite optimisée comme vu précédemment.

### 5.3.3 Stratégie d'exploration

Dans la méthode itérative précédente, il faut choisir à chaque itération une proposition intéressante à fusionner. Ce choix est particulièrement critique car il détermine l'espace des solutions qui peuvent être explorées durant l'optimisation. Même si l'étape de fusion binaire trouve un minimum global car la fonction à minimiser est sous-modulaire, cela n'implique en rien que l'itération des fusions arrive à converger vers un minimum même local de la fonction multi-labels. En effet, c'est seulement un minimum local dans l'espace des solutions explorées qui sera obtenu, d'où l'importance de la stratégie d'exploration. La façon de choisir les propositions va aussi déterminer si les sous-problèmes binaires sont sous-modulaires ou pas.

#### 5.3.3.1 $\alpha$ -expansion

C'est la stratégie d'exploration dites par  $\alpha$ -expansion qui semble la plus efficace jusqu'à présent. La méthode par  $\alpha$ -expansion consiste à effectuer une succession de fusions, avec comme proposition  $\mathbf{L}^p$  un vecteur de valeurs identiques. Les itérations sont effectuées de façon cyclique sur les différentes valeurs de labels possibles jusqu'à stabilisation (BOYKOV, VEKSLER et ZABIH, 2001).

On montre, comme par exemple dans (PAGET, TAREL et L. CARAFFA, 2015), qu'avec l' $\alpha$ -expansion, les sous-problèmes seront toujours sous-modulaires dès que l'énergie (5.1) contient au maximum des termes à deux variables et que la fonction réelle associée à ces termes est symétrique et concave pour les valeurs positives. Ainsi, chaque sous-problème binaire peut être optimisé de façon globale par une recherche de coupe de graphe minimale, ce qui permet d'assigner la totalité des labels (voir la partie 5.2.7.2). Si le minimum global de chaque sous-problème binaire est trouvé lors de la fusion, c'est seulement un minimum local de l'énergie multi-labels qui sera obtenu dans l'espace de recherche des  $\alpha$ -expansions, nommé  $\alpha$ -expansion move (BOYKOV, VEKSLER et ZABIH, 2001).

Il faut noter que l'algorithme  $\alpha$ -expansion original n'est pas introduit comme nous venons de le faire avec l'aide de la programmation pseudo-booléenne. Cette façon de l'introduire permet de dériver une implantation originale et plus facile à programmer que la méthode  $\alpha$ -expansion d'origine.

### 5.3.3.2 $\beta$ -jump

Une autre stratégie d'exploration dites par  $\beta$ -jump est possible même si elle n'est pas aussi efficace en pratique. La méthode par  $\beta$ -jump consiste à fusionner la solution courante avec une proposition qui est la solution courante décalée d'un même incrément des labels. L'intérêt du  $\beta$ -jump réside dans la condition suffisante que doit vérifier l'énergie pour conduire à des sous-problèmes sous-modulaires. Cette condition suffisante est que l'énergie contient au maximum des termes à deux variables et que la fonction réelle associée à ces termes est convexe, voir (PAGET, TAREL et L. CARAFFA, 2015).

### 5.3.3.3 Alternée

Les conditions sur l'énergie avec  $\alpha$ -expansion et  $\beta$ -jump étant assez complémentaires, il est possible d'alterner les deux méthodes afin de traiter des énergies qui ne vérifient aucune de ces deux conditions, comme cela a été proposé dans (PAGET, TAREL et L. CARAFFA, 2015). Alternier  $\alpha$ -expansion et  $\beta$ -jump apparaît comme une approche alternative à l'utilisation de la *roof duality*.

**Exemple 5.6.** Soit l'énergie suivante :

$$f_6(l_1, l_2, l_3) = (l_1 - 2)^2 + (l_2 - 3)^2 + (l_3 - 2)^2 + \lambda|l_1 - l_2| + \lambda|l_2 - l_3|. \quad (5.38)$$

Cette énergie correspond à un problème de débruitage d'une image à  $3 \times 1$  pixels de valeurs d'intensité  $(2, 3, 2)$  avec un terme de régularisation de type  $L_1$  (ou variation totale) entre paires de voisins connexes. Les intensités prennent leurs valeurs ou labels dans l'intervalle des entiers  $[0, 4]$ . On cherche à optimiser cette énergie par  $\alpha$ -expansion à partir de la solution initiale  $\mathbf{L}^0 = (1, 3, 1)$ . Commençons par faire une expansion avec  $\alpha = 2$ . La solution proposée est donc  $\mathbf{L}^p = (2, 2, 2)$ . On constate que la combinaison booléenne auxiliaire peut être écrite comme  $\mathbf{L}(\mathbf{x}) = (2 - \bar{x}_1, 2 + \bar{x}_2, 2 - \bar{x}_3)$ . Le sous-problème binaire est donc après substitution de  $\mathbf{L}(\mathbf{x})$  dans  $f_6$  :

$$\begin{aligned} f_6(\mathbf{L}(\mathbf{x})) &= (\bar{x}_1)^2 + (\bar{x}_2 - 1)^2 + (\bar{x}_3)^2 + \lambda|\bar{x}_1 + \bar{x}_2| + \lambda|\bar{x}_2 + \bar{x}_3| \\ &= \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \lambda(\bar{x}_1 + \bar{x}_2 + \bar{x}_2 + \bar{x}_3) \\ &= 1 + (1 + \lambda)\bar{x}_1 + (2\lambda - 1)\bar{x}_2 + (1 + \lambda)\bar{x}_3 \end{aligned}$$

après simplifications en utilisant le fait que le carré d'un variable binaire est égale à la variable et que la valeur absolue sur une valeur positive s'élimine. On déduit de la dernière expression que le minimum du sous-problème binaire est atteint pour  $\mathbf{x}^* = (1, 1, 1)$ . Le vecteur de labels sélectionné est donc  $(2, 2, 2)$ . En effectuant les autres étapes de l' $\alpha$ -expansion, on verra que cette solution est stable. C'est donc la solution finale qui sera obtenue par  $\alpha$ -expansion. Elle est d'énergie 1 et dans ce cas là, elle est de valeur minimale pour  $f_6$ .

## 5.4 Exemples d'utilisation

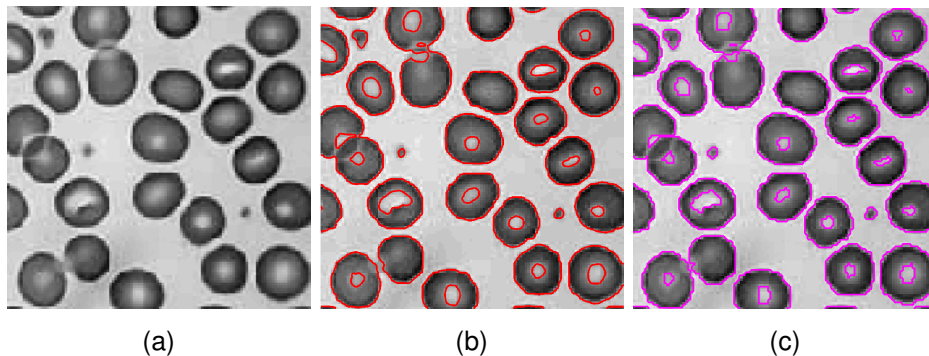
Afin d'illustrer l'intérêt des méthodes d'optimisation présentées dans ce chapitre, nous allons décrire trois applications, leur modèle, et la façon de les optimiser. Le premier exemple est celui de la segmentation binaire d'une image à mettre en relation avec les méthodes décrites dans le chapitre précédent. Ce problème est binaire et sous-modulaire. Le deuxième exemple est celui de la reconstruction 3D à partir de paires stéréoscopiques. Le problème est multi-labels et conduit à des sous-problèmes sous-modulaires. Enfin, le troisième et dernier exemple est le débruitage. Le problème est multi-labels et conduit à des problèmes non sous-modulaires. D'autres exemples sont décrits, par exemple, dans (L. CARAFFA et TAREL, 2013a ; L. CARAFFA et TAREL, 2013b ; Laurent CARAFFA, 2013).

### 5.4.1 Segmentation binaire d'une image

Le problème de la segmentation binaire d'une image a été traité dans le chapitre précédent avec une approche continue. Il est aussi possible de traiter de ce problème de façon discrète par coupe de graphe.

**Figure 5.4**

L'image de la figure 4.10(a) est segmentée avec une méthode d'optimisation continue dite de Split-Bregman (b) et par un méthode discrète par coupe de graphe (c). Les mêmes valeurs de moyennes de région sont utilisés :  $\mu_{int} = 205$  et  $\mu_{ext} = 86$ . Les deux résultats sont assez proches. Le temps de calcul par coupe de graphe est de  $40ms$  ( $\nu = 800$ ).



Ainsi, pour l'image  $I_{ij}, (i, j) \in \Omega$ , utilisée dans la figure 4.10, nous savons qu'elle peut-être segmentée en deux régions d'intensité moyenne  $\mu_{ext} = 86$  et  $\mu_{int} = 205$ . L'énergie discrète à minimiser est donc à partir de (4.17) :

$$f_7(\mathbf{x}) = \sum_{(i,j) \in \Omega} (I_{ij} - \mu_{int})^2 x_{ij} + (I_{ij} - \mu_{ext})^2 \bar{x}_{ij} + \nu |x_{ij} - x_{i+1,j}| + \nu |x_{ij} - x_{i,j+1}|. \quad (5.39)$$

Du fait de l'égalité binaire  $|x - y| = x\bar{y} + y\bar{x}$ , la fonction précédente est sous-modulaire et peut être minimisée globalement. Le résultat est montré en

figure 5.4(c) et comparé au résultat de la méthode dite Split-Bergman qui donne aussi la solution au minimum global (figure 5.4(b)).

## 5.4.2 Reconstruction stéréoscopique

La reconstruction à partir de paires stéréoscopique consiste à retrouver le modèle 3D de la scène à partir de deux vues dont l'une est décalée horizontalement sur le plan focal par rapport à l'autre. Grâce à cette disposition, dite à géométrie épipolaire rectifiée, la projection d'un point de la scène 3D dans les deux caméras se fera sur la même ligne horizontale. Si l'on connaît cette projection dans les deux images, alors, il est possible, par triangulation, de retrouver la profondeur. La différence de colonne d'une projection entre les deux images s'appelle la disparité. L'ensemble des disparités d'une image est appelée la carte de disparité.

Supposons que nous avons deux images rectifiées  $I^g$  et  $I^d$  de la même scène. La reconstruction stéréoscopique consiste à estimer la carte de disparité  $D$  à partir des deux images. L'ensemble des pixels est noté  $\Omega$  et est indicé par le couple d'indices  $(i, j)$ .

L'énergie du modèle dans le cas le plus simple est la suivante :

$$f_8(D) = \sum_{(i,j) \in \Omega} (I_{ij}^g - I_{i(j+D_{ij})}^d)^2 + \nu |D_{ij} - D_{i(j+1)}| + \nu |D_{ij} - D_{(i+1)j}|. \quad (5.40)$$

Le premier terme de l'énergie représente le terme d'attache aux données, c'est à dire l'appariement entre l'image gauche  $I^g$  et l'image droite  $I^d$ . C'est simplement la différence de leurs intensités sachant la disparité. Quand l'intensité de l'image gauche est proche de celle de l'image droite, le score est faible. Inversement, quand les deux intensités sont différentes, le score est fort. Le second terme de l'énergie est un terme de régularisation dit  $L_1$  ou variation totale qui a pour but de favoriser l'égalité des disparités entre pixels voisins.

Nous cherchons à optimiser l'énergie  $f_8$  par rapport à la variable discrète  $D$ . L'avantage de cette énergie est son caractère sous-modulaire. En effet, le terme d'attache aux données est unaire, c'est à dire que chaque terme est fonction d'une seule variable de  $D_{ij}$ . Le terme de régularisation est fonction de deux variables, paire et concave pour les valeurs positives. En conséquence, l'énergie (5.40) conduit toujours à des sous-problèmes sous-modulaires quand on utilise  $\alpha$ -expansion. Le terme de régularisation est aussi convexe, donc l'énergie (5.40) conduit toujours à des sous-problèmes sous-modulaires quand on utilise  $\beta$ -jump.

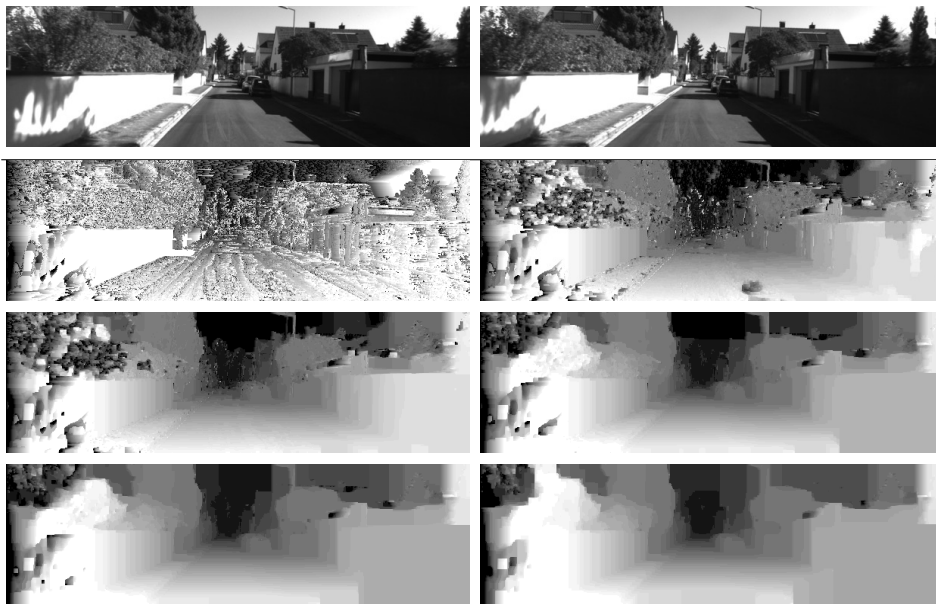
La figure 5.5 montre la solution minimale de l'énergie (5.40) avec l'algorithme  $\alpha$ -expansion pour différentes valeurs de  $\nu$  sur une paire stéréoscopique de la base *KITTI Vision Benchmark Suite*<sup>1</sup>. Nous pouvons voir que lorsque  $\nu = 0$ , c'est à dire sans régularisation, le résultat n'est pas bon à cause d'ambiguïtés. Avec  $\nu = 1$ , la carte de disparité obtenue est plus lisse. Il reste cependant quelques zones comme le bas de la chaussée mal reconstruites. Avec  $\nu = 4$ , la chaussée

1. <http://www.cvlibs.net/datasets/kitti/>

est plus lisse et donc mieux reconstruite. Avec  $\nu = 16$ , la chaussée au loin est encore mieux reconstruite, cependant, un plan fronto-parallèle s'est formé au niveau de bord droit dû au fort poids du terme de régularisation. Ce phénomène est encore plus important avec  $\nu = 32$  et  $\nu = 64$ . Pour cette scène,  $\nu = 4$  semble donc être un bon compromis.

**Figure 5.5**

Résultat de reconstruction par  $\alpha$ -expansion sur (5.40) en fonction de la valeur de  $\nu$ . Première ligne : l'image gauche et droite de la scène. Seconde ligne :  $\nu = 0$  et  $\nu = 1$ . Troisième ligne :  $\nu = 4$  et  $\nu = 16$ . Quatrième ligne :  $\nu = 32$  et  $\nu = 64$ .



Pour palier au problème de la reconstruction de la chaussée, il est possible de faire une étape de fusion entre la carte de disparité obtenue par minimisation de l'énergie (5.40) et une carte de disparité consistant au plan de la route. Le résultat de cette fusion est montré sur une autre paire d'images plus faciles à comprendre en figure 5.6. Nous constatons que les profondeurs sur les objets fronto-parallèles sont bien préservées et que la route est mieux reconstruite.

**5.4.3 Débruitage d'une image**

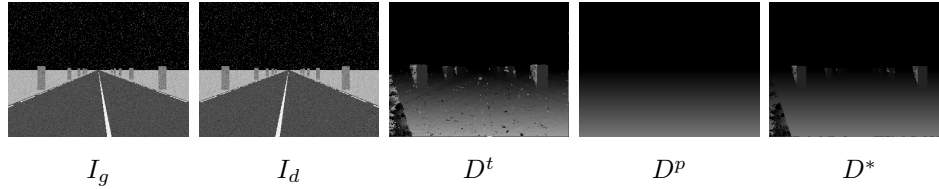
Le problème du débruitage d'une image consiste à enlever le bruit en chaque pixel. Cela peut se faire par lissage des zones uniformes en préservant les bords importants.

Étant donné une image bruitée  $I$ , et dans le cas d'un bruit supposé additif, indépendant et gaussien, le problème du débruitage peut être posé comme la minimisation de l'énergie suivante par rapport à la variable  $D$  qui représente



**Figure 5.6**

Reconstruction raffinée par fusion avec un modèle a priori de la route. De gauche à droite : l'image gauche et droite, la carte de disparité obtenue par la minimisation de (5.40), la carte de disparité de la route et le résultat de la fusion.



l'image débruitée :

$$f_9(D) = \sum_{(i,j) \in \Omega} (I_{ij} - D_{ij})^2 + \lambda g(|D_{ij} - D_{i(j+1)}|) + \lambda g(|D_{ij} - D_{(i+1)j}|) \quad (5.41)$$

où  $g$  est une fonction croissante sur les valeurs positives. Le premier terme est le terme d'attache aux données qui empêche l'image  $D$  de s'éloigner trop de l'image bruitée  $I$ . Les deux derniers termes sont des termes de régularisation qui force  $D$  à être lisse. Lorsque  $g$  est l'identité, on retrouve à nouveau la régularisation  $L_1$  ou variation totale. Si ce choix est assez efficace en terme de lissage, il a le défaut de faire apparaître des marches d'escaliers sur les zones où l'intensité varie de façon assez lente comme par exemple sur la joue de Lena dans la figure 5.7. Il est donc préférable d'utiliser pour  $g$  une fonction  $L_1$  en modifiant la forme en zéro pour avoir une petite cuvette quadratique au lieu d'un angle droit. Cela permet d'obtenir des dégradés plus lisses comme illustré en figure 5.7. La fonction n'étant plus concave, il faut utiliser la méthode alternée pour résoudre chaque sous-problème binaire.

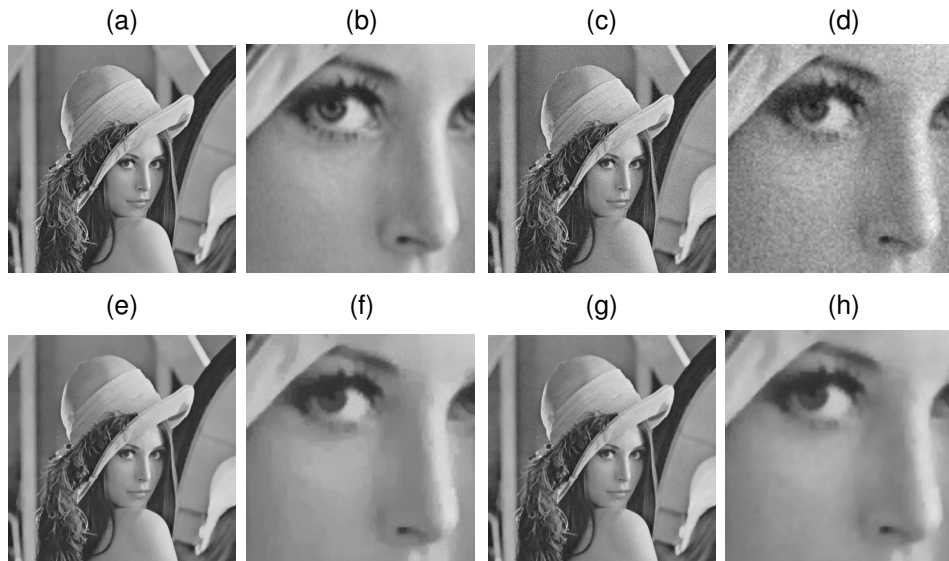
## 5.5 Conclusion

Le cadre de l'optimisation des fonctions pseudo-booléennes a fait apparaître l'importance de la distinction entre fonction non sous-modulaire et fonction sous-modulaire. Dans ce dernier cas, des algorithmes en temps polynomial permettent de trouver un minimum global de l'énergie par exemple par coupe de graphe.

Mais, l'ensemble des fonctions sous-modulaires est assez restreint. En effet, dans le cadre binaire, seuls les énergies avec des termes d'ordres au maximum 2 à coefficients négatifs sont sous-modulaires et il est alors possible de les optimiser globalement en un temps polynomial. Les fonctions binaires avec des termes d'ordre supérieur à 2 ne sont généralement pas sous-modulaires. Il existe des techniques qui permettent de transformer une fonction pseudo-booléenne quelconque en une fonction pseudo-booléenne quadratique partageant le même minimum global. Mais ces techniques de transformation entraînent l'apparition de termes non sous-modulaires. Il n'est donc plus possible d'appliquer les algorithmes d'optimisation globale sous-modulaire.

**Figure 5.7**

Image originale (a) et un zoom (b). La même image bruitée par un bruit gaussien (c) et son zoom (d). L'image débruitée avec une régularisation  $L_1$  (e) et son zoom (f). Enfin, l'image débruitée avec une régularisation  $L_1$  modifiée par une petite cuvette en zéro (g) et son zoom (h). Le débruitage avec  $L_1$  est obtenu par  $\alpha$ -expansion et par une méthode alternant  $\alpha$ -expansion et  $\beta$ -jump dans le cas  $L_1$  modifiée.



Étonnamment, et malgré le caractère non sous-modulaire de la fonction à minimiser, certaines affectations de labels, appartenant au minimum global, peuvent néanmoins être trouvées en un temps polynomial. En effet, la structure épaisse de certains graphes permet de détecter, pour un sous-ensemble de variables, si l'affectation d'un label pour une variable donnée appartient au minimum global ou non. Cette propriété est importante car elle permet, même si le minimum global d'une énergie ne peut pas être atteint complètement du fait de sa non sous-modularité, d'identifier un sous-ensemble de variables pour lesquelles on peut trouver l'affectation appartenant au minimum global. La méthode pour le faire est la *roof duality*.

Pour passer à l'optimisation multi-labels, une heuristique est de décomposer le problème en une succession de sous-problèmes binaires, nommés fusions, où il faut choisir entre une solution courante, et une proposition de label. Chaque étape de fusion sélectionne une solution globale de l'énergie du sous-problème binaire, si elle est sous-modulaire. Pour autant, en itérant les fusions, il n'y a pas de garantie de converger vers un minimum local.

Enfin, nous avons vu que l'optimisation de fonctions pseudo-booléennes quadratiques apporte un cadre assez large pour optimiser les énergies rencontrées en traitement d'image et peut être aussi utilisée dans d'autres contextes.

Il faut noter que cette présentation succincte ne représente qu'une petite partie de l'optimisation des fonctions pseudo-booléennes. Pour aller plus loin, l'article (BOROS, Peter L. HAMMER et TAVARES, 2006) propose de nouvelles méthodes pour l'optimisation de fonctions pseudo-booléennes quadratiques tel que le *probing* qui permet d'améliorer le calcul de la *roof duality*. Autre exemple, la thèse (STRANDMARK, 2012), soutenue en 2012, propose un état de l'art complet sur l'optimisation discrète appliquée au domaine de la vision par ordinateur.

### 5.5.1 Récapitulatif

Pour résumer, les points clés à retenir de l'optimisation des fonctions pseudo-booléennes sont :

- N'importe quelle fonction pseudo-booléenne peut être transformée en une posiforme.
  - Les affectations vérifiant la persistance forte avec la *roof duality* peuvent être trouvées en un temps polynomial par poussage de flot.
  - Les affectations persistantes faibles peuvent être trouvées en résolvant la posiforme réduite.
- Si la fonction pseudo-booléenne est sous-modulaire, alors on peut trouver un minimum global en un temps polynomial.
- Une fonction pseudo-booléenne quelconque peut toujours être réduite en une posiforme quadratique en ajoutant des variables supplémentaires, et ayant le même minimum global.
  - Il existe plusieurs réductions possibles.
  - La réduction entraîne l'apparition de termes non sous-modulaires.
  - Le nombre et les coefficients des termes non sous-modulaires dépendent directement de la réduction utilisée.
  - Le nombre et les coefficients des termes non sous-modulaires influent sur le nombre d'affectations persistantes fortes.
- On peut décomposer un problème multi-labels en une série de sous-problèmes binaires en utilisant la fusion et diverses stratégies d'exploration ( $\alpha$ -expansion,  $\beta$ -jump, alternée), mais on n'obtient alors pas forcément un minimum global.

### 5.5.2 Outils libres

Plusieurs implantations d'algorithmes sous licence libre sont disponibles pour résoudre les différents problèmes abordés dans ce chapitre :

- $\alpha$ -expansion : <http://vision.csd.uwo.ca/code/>
- Optimisation de fonction pseudo-booléenne quadratique : <http://pub.ist.ac.at/~vnk/software.html>
- Réduction de fonction pseudo-booléennes quelconques en quadratique : <http://www.f.waseda.jp/hfs/software.html>



## Bibliographie

- [BH02] Endre BOROS et Peter L HAMMER. « Pseudo-boolean optimization ». In : *Discrete applied mathematics* 123.1 (2002), p. 155-225.
- [BHT06] Endre BOROS, Peter L. HAMMER et Gabriel TAVARES. *Preprocessing of unconstrained quadratic binary optimization*. Rapp. tech. Rutgers University, 2006.
- [BVZ01] Yuri BOYKOV, Olga VEKSLER et Ramin ZABIH. « Fast Approximate Energy Minimization via Graph Cuts ». In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23 (11 nov. 2001), p. 1222-1239.
- [Car13] Laurent CARAFFA. « Reconstruction 3D à partir de paires stéréoscopiques en conditions dégradées ». Thèse de doct. Université Pierre et Marie Curie, 2013.
- [CT13a] L. CARAFFA et J.-P. TAREL. « Markov Random Field Model for Single Image Defogging ». In : *Proceedings of IEEE Intelligent Vehicle Symposium (IV'2013)*. <http://perso.lcpc.fr/tarel.jean-philippe/publis/iv13.html>. Gold Coast, Australia, 2013, p. 994-999.
- [CT13b] L. CARAFFA et J.-P. TAREL. « Stereo Reconstruction and Contrast Restoration in Daytime Fog ». In : *Proceedings of Asian Conference on Computer Vision (ACCV'12)*. T. IV. LNCS. <http://perso.lcpc.fr/tarel.jean-philippe/publis/accv12.html>. Daejeon, Korea : Springer, 2013, p. 13-25.
- [Fix+11] Alexander FIX et al. « A graph cut algorithm for higher-order Markov Random Fields ». In : *Proceedings of the 2011 International Conference on Computer Vision. ICCV '11*. Washington, DC, USA : IEEE Computer Society, 2011, p. 1020-1027. ISBN : 978-1-4577-1101-5. DOI : 10.1109/ICCV.2011.6126347. URL : <http://dx.doi.org/10.1109/ICCV.2011.6126347>.
- [GBP11] A. C. GALLAGHER, D. BATRA et D. PARIKH. « Inference for order reduction in Markov random fields ». In : *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '11*. Washington, DC, USA : IEEE Computer Society, 2011, p. 1857-1864. ISBN : 978-1-4577-0394-2. DOI : 10.1109/CVPR.2011.5995452. URL : <http://dx.doi.org/10.1109/CVPR.2011.5995452>.

Optimisation de formes en sciences de l'ingénieur

- [Ish03] Hiroshi ISHIKAWA. « Exact optimization for Markov random fields with convex priors ». In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.10 (2003), p. 1333-1336.
- [Ish09] Hiroshi ISHIKAWA. « Higher-order clique reduction in binary graph cut ». In : *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE, 2009, p. 2993-3000. ISBN : 978-1-4244-3992-8. DOI : <http://doi.ieeecomputersociety.org/10.1109/CVPRW.2009.5206689>.
- [Ish11] Hiroshi ISHIKAWA. « Transformation of General Binary MRF Minimization to the First-Order Case ». In : *IEEE Trans. Pattern Anal. Mach. Intell.* 33.6 (2011), p. 1234-1249.
- [KS11] Fredrik KAHL et Petter STRANDMARK. « Generalized roof duality for pseudo-boolean optimization ». In : *Proceedings of the 2011 International Conference on Computer Vision. ICCV '11*. Washington, DC, USA : IEEE Computer Society, 2011, p. 255-262. ISBN : 978-1-4577-1101-5. DOI : 10.1109/ICCV.2011.6126250. URL : <http://dx.doi.org/10.1109/ICCV.2011.6126250>.
- [KZ02] Vladimir KOLMOGOROV et Ramin ZABIH. « What Energy Functions Can Be Minimized via Graph Cuts? » In : *Proceedings of the 7th European Conference on Computer Vision-Part III. ECCV '02*. London, UK, UK : Springer-Verlag, 2002, p. 65-81. ISBN : 3-540-43746-0. URL : <http://dl.acm.org/citation.cfm?id=645317.649315>.
- [Lem+10] Victor LEMPITSKY et al. « Fusion Moves for Markov Random Field Optimization ». In : *IEEE Trans. Pattern Anal. Mach. Intell.* 32.8 (août 2010), p. 1392-1405. ISSN : 0162-8828. DOI : 10.1109/TPAMI.2009.143. URL : <http://dx.doi.org/10.1109/TPAMI.2009.143>.
- [PTC15] M. PAGET, J.-P. TAREL et L. CARAFFA. « Extending  $\alpha$ -expansion to a larger set of regularization functions ». In : *Proceedings of International Conference on Image Processing (ICIP'15)*. <http://perso.lcpc.fr/tarel.jean-philippe/publis/icip15.html>. Quebec City, Canada, 2015.
- [Ros75] I. ROSENBERG. « Reduction of bivalent maximization to the quadratic case. » In : *Cahiers du Centre d'Etudes de Recherche Operationnelle* (1975), 17 :7174.
- [Str12] Petter STRANDMARK. « Discrete Optimization in Early Vision ». Thèse de doct. Centre for Mathematical Sciences LTH, Lund University, Sweden, 2012.