



HAL
open science

Programmation mathématique en tomographie discrète

Ghassen Tlig

► **To cite this version:**

Ghassen Tlig. Programmation mathématique en tomographie discrète. Traitement des images [eess.IV]. Conservatoire national des arts et métiers - CNAM; Université Tunis El Manar. Faculté des Sciences Mathématiques, Physiques et Naturelles de Tunis (Tunisie), 2013. Français. NNT : 2013CNAM0886 . tel-00957445

HAL Id: tel-00957445

<https://theses.hal.science/tel-00957445>

Submitted on 10 Mar 2014

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

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



le cnam

École Doctorale d'Informatique, Télécommunications et Électrique
Centre d'Étude et De Recherche en Informatique et Communications (EA 4629)
Faculté des Sciences Mathématiques, Physiques et Naturelles de Tunis

THÈSE présentée par : **Ghassen TLIG**

soutenue le : **13 novembre 2013**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Discipline/ Spécialité : Informatique

Programmation Mathématique en Tomographie Discrète

THÈSE dirigée par :

Alain BILLIONNET, co-directeur de thèse

Professeur à l'Ecole Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise

Ezzeddine ZAGROUBA, co-directeur de thèse

Professeur à la Faculté des Sciences Mathématiques, Physiques et Naturelles de Tunis

Fethi JARRAY, co-encadrant

Maitre-assistant à l'Institut Supérieur d'Informatique de Medenine

RAPPORTEURS :

Jacques CARLIER, Professeur à l'Université de Technologie de Compiègne

Laurent VUILLON, Professeur à l'Université de Savoie

JURY :

Christophe PICOULEAU, Professeur au Conservatoire National des Arts et Métiers

Frédéric ROUPIN, Professeur à l'Université Paris 13

Remerciements

À l'issue de ce travail, je tiens à remercier vivement :

- Alain Billionnet de m'avoir donné l'opportunité de faire cette thèse. Il a su poser les bonnes questions qui m'ont permis de familiariser avec le domaine.
- Ezzedine Zagrouba pour l'attention qu'il a portée à mes travaux et la confiance qu'il m'a accordée m'ont bien aidé à progresser au long de ce travail.
- Fethi Jarray pour l'attention constante qu'il a portée à mes travaux depuis mon stage de Maîtrise jusqu'à la phase de rédaction de cette thèse et aux nombreuses relectures et discussions intéressantes qui l'accompagnent.

Je souhaite adresser mes sincères remerciements à l'ensemble des membres de mon jury de thèse : Pr. Jacques Carlier, Pr. Laurent Vuillon, Pr. Christophe Picouveau et Pr. Frédéric Roupin, qui ont accepté d'examiner ma thèse.

Je voudrais remercier ensuite tous les membres de laboratoire CEDRIC pour l'accueil constamment chaleureux malgré mes passages toujours trop courts, en particulier les membres de l'équipe OC et mes collègues thésards à qui je souhaite un bon courage.

Je tiens à remercier particulièrement les membres de l'équipe SIIVA du l'institut supérieur d'informatique de tunis et tous mes collègues enseignants à l'institut supérieur d'informatique de mednine.

Je remercie aussi Christophe Picouveau pour l'aide et les conseils qu'il m'a donnés.

Enfin, je remercie tous les amis qui m'ont aidé à des titres divers à mener ce travail à bien. Ils sont heureusement trop nombreux pour être cités ici sans en oublier aucun.

Résumé

La tomographie est un ensemble de techniques visant à reconstruire l'intérieur d'un objet sans toucher l'objet lui-même comme dans le cas d'un scanner. Les principes théoriques de la tomographie ont été énoncés par Radon en 1917. On peut assimiler l'objet à reconstruire à une image, matrice, etc.

Le problème de reconstruction tomographique consiste à estimer l'objet à partir d'un ensemble des projections obtenues par mesures expérimentales autour de l'objet à reconstruire. La tomographie discrète étudie le cas où le nombre de projections est limité et l'objet est défini de façon discrète. Les champs d'applications de la tomographie discrète sont nombreux et variés. Citons par exemple les applications de type non destructif comme l'imagerie médicale. Il existe d'autres applications de la tomographie discrète, comme les problèmes d'emplois du temps.

La tomographie discrète peut être considérée comme un problème d'optimisation combinatoire car le domaine de reconstruction est discret et le nombre de projections est fini. La programmation mathématique en nombres entiers constitue un outil pour traiter les problèmes d'optimisation combinatoire.

L'objectif de cette thèse est d'étudier et d'utiliser les techniques d'optimisation combinatoire pour résoudre les problèmes de tomographie ¹.

Mots-Clés : Tomographie discrète, Reconstruction d'images, Optimisation Combinatoire, Programmation mathématique, Programmation linéaire, Linéarisation et Convexification, Métaheuristiques.

1. Les résultats obtenus montrent que la programmation mathématique est une approche intéressante pour ce domaine.

Abstract

The tomographic imaging problem deals with reconstructing an object from a data called a projections and collected by illuminating the object from many different directions. A projection means the information derived from the transmitted energies, when an object is illuminated from a particular angle. The solution to the problem of how to reconstruct an object from its projections dates to 1917 by Radon. The tomographic reconstructing is applicable in many interesting contexts such as nondestructive testing, image processing, electron microscopy, data security, industrial tomography and material sciences.

Discrete tomography (DT) deals with the reconstruction of discrete object from limited number of projections. The projections are the sums along few angles of the object to be reconstruct. One of the main problems in DT is the reconstruction of binary matrices from two projections. In general, the reconstruction of binary matrices from a small number of projections is undetermined and the number of solutions can be very large. Moreover, the projections data and the prior knowledge about the object to reconstruct are not sufficient to determine a unique solution. So DT is usually reduced to an optimization problem to select the best solution in a certain sense.

In this thesis, we deal with the tomographic reconstruction of binary and colored images. In particular, research objectives are to derive the combinatorial optimization techniques in discrete tomography problems.

Keywords : Discrete Tomography, Reconstructing Image, Combinatorial Optimization, Integer programming, Convexification, Linearization.

Table des matières

Table des matières	vii
Table des figures	xi
Liste des tableaux	xiii
1 Introduction Générale	1
2 Problème de tomographie discrète	3
2.1 Introduction	4
2.2 Problème standard et applications	4
2.2.1 Définitions	5
2.2.2 Reconstruction d'une matrice binaire	6
2.2.3 Applications	8
2.2.3.1 Imagerie médicale	8
2.2.3.2 Microscopie électronique	8
2.2.3.3 Compression de données	9
2.2.3.4 Géométrie convexe	9
2.2.3.5 Physique statistique	9
2.2.3.6 Planification de personnel	9
2.2.3.7 Radio Tomographic Imaging (RTI)	10
2.3 Méthodes de reconstruction	10
2.3.1 Théorie des graphes	10
2.3.2 Programmation mathématique	10
2.3.3 Techniques d'optimisation combinatoire	11

TABLE DES MATIÈRES

2.4	Problèmes traités en tomographie discrète	12
2.4.1	Reconstruction de matrices convexes	12
2.4.1.1	Reconstruction d'une matrice binaire hv-convexe	15
2.4.2	Approximations et reconstruction d'un hv-convexe	15
2.4.2.1	Relaxation lagrangienne pour la reconstruction de matrice binaire hv-convexe	16
2.4.2.2	Reconstruction d'un polyomino hv-convexe	20
2.4.3	Reconstruction de matrices périodiques	20
2.4.4	Reconstruction de matrices adjacentes	21
2.4.5	Reconstruction de matrices colorées	23
2.5	Conclusion	23
3	Programmation mathématique pour la reconstruction des images	26
3.1	Introduction	27
3.2	Reconstruction d'une image binaire convexe	27
3.3	Programmation quadratique en 0-1 pour reconstruire une image binaire hv-convexe	27
3.3.1	Linéarisation du programme QP	29
3.3.2	Convexification du programme QP	31
3.4	Système algébrique pour reconstruire une image binaire convexe	34
3.5	Reconstruction avec variables fixées	35
3.6	Résultats numériques	36
3.6.1	Interprétations	37
3.7	Conclusion	41
4	Résolution de cas polynomiaux	43
4.1	Introduction	43
4.2	Reconstruction polynomiale d'une image binaire convexe	44
4.2.1	Reconstruction à partir de deux projections constantes	44
4.2.2	Reconstruction à partir d'une seule projection constante	47
4.2.2.1	Reconstruction d'un polyomino hv-convexe	47
4.2.2.2	Reconstruction d'un hv-convexe	49

TABLE DES MATIÈRES

4.3	Reconstruction d'une image binaire h-convexe à partir d'une projection verticale	50
4.4	Reconstruction d'une image binaire hv-convexe avec projection bornée	52
4.5	Conclusion	53
5	Métaheuristiques pour la reconstruction des images binaires	56
5.1	Introduction	57
5.2	Reconstruction des images binaires hv-convexes	57
5.2.1	Modélisation du problème comme un problème de recherche	58
5.2.1.1	Fonction objectif	58
5.2.1.2	Solution initiale.	59
5.2.1.3	Voisinage	59
5.2.2	Méthode de recuit simulé	60
5.2.3	Paramètres de recuit simulé	61
5.3	Résultats numériques	62
5.3.1	Reconstruction $Image_{hv-conv}(H, V)$	62
5.3.2	Interprétations	63
5.4	Conclusion	65
6	Reconstruction des images colorées	67
6.1	Introduction	68
6.2	Définition du problème	68
6.3	Programmation entière pour la reconstruction d'une image colorée	70
6.4	Méthode de recuit simulé pour la reconstruction des images colorées	73
6.4.1	Fonction objectif	73
6.4.2	Solution initiale	74
6.4.3	Notion de voisinage	75
6.4.4	Algorithme de recuit simulé	77
6.5	Résultats numériques	78
6.5.1	Interprétations	79
6.6	Conclusion	79
7	Conclusion Générale	86

TABLE DES MATIÈRES

Références	90
Bibliographie de l'auteur	95

Table des figures

2.1	Une matrice binaire avec ses projections horizontales H , verticales V , diagonales D et anti-diagonale A	5
2.2	Définition de q – <i>adjacence</i> : de gauche à droite $q \in \{2, 4, 6, 8\}$	6
2.3	De gauche à droite un chemin 4-connexe, un chemin 8-connexe et une matrice binaire contient deux composantes 4-connexe ou une composante 8-connexe	7
2.4	Matrice binaire avec ses projections orthogonales et le graphe orienté correspondant	11
2.5	De gauche à droite : $\{h, v, hv, hvd, hvda\}$ -convexe	13
2.6	A gauche un Q -convexe et à droite un non Q -convexe : $P \notin A$ alors que $R_1(P) \cap A \neq 0$ et $R_4(P) \cap A \neq 0$	14
2.7	La construction d’un graphe pour PRL	19
2.8	De gauche à droite une matrice binaire (2,3)-périodique et (2,3)-alternée	21
2.9	Contrainte d’adjacence p – q – <i>adjacence</i> : de gauche à droite $q = \{2, 4, 6, 8\}$	21
2.10	Matrice 2-colorée avec ses projections horizontales et verticales pour chaque couleur.	24
3.1	Exemple de reconstruction d’une image binaire convexe	37
4.1	Bloc de taille $k_2 \times k_1$ sur les lignes $\{1, \dots, k_2\}$ et les colonnes $\{1, \dots, k_1\}$	45
4.2	Solution du problème hv – <i>conv</i> (H^{k_1}, V^{k_2}) est composée de $r = m/k_2$ blocs	46

TABLE DES FIGURES

4.3	Trois lignes r , s et t avec : $jr \leq js \geq jt$	48
4.4	Trois lignes r , s et t avec : $jr \geq js \leq jt$	48
4.5	Deux solutions différentes du problème $phv - conv(H^3, V)$: $H^3 = (3, 3, 3, 3, 3, 3)$ et $V = (1, 1, 3, 4, 5, 3, 1)$	49
4.6	Réduction polynomiale : Instance du problème $hv - conv(H, V)$ et instance de $NMTS(A, B, C)$, avec $A = \{1, 3, 4\}$; $B = \{2, 1, 1\}$; $C = \{3, 4, 5\}$	53
6.1	Une image 3-colorée convexe avec les projections horizontales et verticales de chaque couleur	69
6.2	Une image colorée et le problème du flot max à coût min dans le graphe associé à Q_1 : $G(H, V, E)$; $H^1 = (2, 0, 1)$; $H^2 = (0, 2, 1)$ et $V^1 = (2, 1, 0)$, $V^2 = (1, 1, 1)$	74
6.3	Deux images colorées A (à gauche) et B (à droite) équivalentes : à l'aide d'une suite finie d'opérations élémentaires il est possible de transformer A en B ou inversement	77
6.4	Exemple de reconstruction des images colorées	84

Liste des tableaux

2.1	Complexité du problème de reconstruction d'une matrice binaire T -convexe. '*' : problème ouvert	14
2.2	Résultats de complexité du problème de reconstruction d'une matrice périodique.	21
2.3	Résultats de complexité du problème $p - q - adj(H, V)$	22
3.1	Résultats de reconstruction par les programmes linéaires	38
3.2	Résultats de reconstruction par les programmes convexes	39
3.3	Résultats de reconstruction par LP1 et le système S avec variables fixées; (-) : solution non trouvée au bout d'une heure de calcul . .	40
4.1	Complexité de problème de reconstruction d'une image binaire convexe : $hv - conv(H^k, V^k)$ et $hv - conv(H^k, V)$ (resp. $hv - conv(H, V^k)$).	54
4.2	Complexité de problème de reconstruction d'une image binaire convexe : $h - conv(V, m)$ (resp. $v - conv(H, n)$) et $hv - conv(H, n)$ (resp. $hv - conv(V, n)$)	54
5.1	Résultats de reconstruction pour les images binaires -I-	63
5.2	Résultats de reconstruction pour les images binaires -II-	64
5.3	Résultats de reconstruction pour les images binaires -III-; (-) : solution non trouvée au bout de 900s de calcul	64
6.1	Exemples d'opérations élémentaires pour les images colorées . . .	76
6.2	Résultats numériques pour la reconstruction des images colorées, ensemble 1	80

LISTE DES TABLEAUX

6.3 Résultats numériques pour la reconstruction des images colorées, ensemble 2	81
6.4 Résultats numériques pour la reconstruction des images colorées, ensemble 3	82
6.5 Résultats numériques pour la reconstruction des images colorées, ensemble 4	83

Chapitre 1

Introduction Générale

La tomographie (racine grecque tomê, coupe, section - Graphien : écrire) est une technique qui consiste à reconstruire le volume d'un objet à partir d'une série de mesures effectuées depuis l'extérieur de cet objet. Le résultat est une reconstruction de certaines propriétés de l'intérieur de l'objet. L'utilisation de la tomographie est connue dans le cadre de l'imagerie médicale pour laquelle l'objet à reconstruire est le corps humain. L'intérêt de la tomographie est de pouvoir connaître les caractéristiques de l'objet sans utiliser des outils diagnostiques invasifs et d'apporter un certain confort au patient, lorsqu'il s'agit de réaliser un examen médical.

La reconstruction tomographique est donc bien un problème inverse puisqu'il s'agit d'estimer la structure d'un objet à partir d'un ensemble des mesures effectuées autour de l'objet. De manière générale, il existe deux méthodes de reconstruction tomographique :

- La tomographie de transmission : L'objet étudié est soumis à un rayonnement. Ce rayonnement transmis par l'objet est ensuite récupéré puis traité. Les photons émis par la source subissent des interactions avec l'objet étudié et le rayonnement est atténué de manière plus ou moins intense.
- La tomographie d'émission : L'objet étudié est lui-même l'émetteur de rayonnement. La tomographie d'émission est idéale pour étudier les objets qui rayonnent naturellement.

La tomographie est aussi un problème d'optimisation combinatoire car le domaine de reconstruction est discret et le nombre de projections est fini.

- Résoudre un problème d'optimisation combinatoire consiste à trouver un ensemble optimal, dans le sens de maximiser ou minimiser une certaine fonction objectif, parmi une famille d'ensembles admissibles. De nombreux problèmes et applications en informatique peuvent être formulées à l'aide d'un problème d'optimisation combinatoire. Les problèmes d'optimisation combinatoire sont faciles à définir, mais ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes difficiles et ne possèdent donc pas à ce jour des solutions algorithmiques efficaces valables pour les données de grande taille. Il faut alors faire appel à des outils mathématiques et algorithmiques pour essayer de résoudre, de façon exacte ou approchée, des instances pour un problème donné.
- La programmation mathématique en nombres entiers constitue un outil pour traiter des problèmes d'optimisation combinatoire. Pour un nouveau problème, on commence par le modéliser à l'aide d'un modèle mathématique. On peut trouver plusieurs modèles pour un même problème, dont certains sont meilleurs que d'autres. La phase de modélisation permet d'utiliser les outils et les méthodes de la programmation mathématique, en particulier la programmation linéaire et la programmation linéaire en nombres entiers.

L'objectif principal de cette thèse est d'utiliser les techniques d'optimisation combinatoire et en particulier la programmation mathématique pour résoudre les problèmes de reconstruction tomographiques.

Chapitre 2

Problème de tomographie discrète

Sommaire

2.1	Introduction	4
2.2	Problème standard et applications	4
2.2.1	Définitions	5
2.2.2	Reconstruction d'une matrice binaire	6
2.2.3	Applications	8
2.3	Méthodes de reconstruction	10
2.3.1	Théorie des graphes	10
2.3.2	Programmation mathématique	10
2.3.3	Techniques d'optimisation combinatoire	11
2.4	Problèmes traités en tomographie discrète	12
2.4.1	Reconstruction de matrices convexes	12
2.4.2	Approximations et reconstruction d'un hv-convexe	15
2.4.3	Reconstruction de matrices périodiques	20
2.4.4	Reconstruction de matrices adjacentes	21
2.4.5	Reconstruction de matrices colorées	23
2.5	Conclusion	23

2.1 Introduction

La tomographie est une technique qui consiste à reconstituer la structure d'un objet à partir de ses images radiologiques classiques. Dans le cas continu, on peut reconstruire l'objet grâce à une transformation explicite (la transformation de Radon). Cette technique trouve un équivalent dans le discret : il s'agit de trouver un sous ensemble de \mathbb{Z}^n à partir d'un ensemble de projections.

La tomographie discrète est une nouvelle discipline informatique trouvant ses fondements aussi bien en algorithmique combinatoire qu'en géométrie discrète. Nous pouvons citer parmi ses champs d'applications en informatique : l'imagerie médicale, la cristallographie, la planification de personnel, l'ordonnancement, la compression de données, etc.

La reconstruction de matrices binaires à partir de leurs projections orthogonales est le sujet le plus étudié à ce jour. Le problème peut être formulé de la manière suivante : étant donnés $H = (h_1, h_2, \dots, h_m)$ et $V = (v_1, v_2, \dots, v_n)$ deux vecteurs à coordonnées entières positives, est-il possible de reconstruire une matrice binaire de taille $(m \times n)$ qui respecte les vecteurs (H, V) ? Une généralisation de ce problème consiste à reconstruire des images colorées ou en niveaux de gris. Les projections donnent le nombre de cellules colorées par chaque couleur.

2.2 Problème standard et applications

D'après Maurice Nivat [40] : « La tomographie est l'art de reconstituer une image à partir d'informations fragmentaires et locales et son utilité n'est pas à démontrer, tous les malades qui ont vu leur mal détecté au scanner ou à l'IRM sont là pour en témoigner. L'histoire assez longue de la tomographie à usage médical est fascinante par le mélange de difficultés purement mathématiques et d'innovations technologiques qui la constituent depuis pas loin d'un siècle. La tomographie discrète, elle est beaucoup plus récente puisque le terme même a été employé pour la 1ère fois en 1994. L'image est alors constituée de pixels et les données sont les sommes pondérées des pixels appartenant à des parties de

2. Problème de tomographie discrète

l'image. Pourtant les premiers résultats remontent aux années 50, ce sont ceux de Ryser sur l'existence de matrices à coefficients 0 ou 1 ayant des sommes données en lignes et en colonnes.»

2.2.1 Définitions

Nous notons par

- $a_{i,j}$: les éléments d'une matrice binaire A de taille (m, n) , $1 \leq i \leq m$ et $1 \leq j \leq n$.
- $h_i = \sum_{j=1}^n a_{i,j}$: la projection horizontale de la ligne i , $1 \leq i \leq m$.
- $H = (h_1, h_2, \dots, h_m)$: la projection horizontale de la matrice A .
- $v_j = \sum_{i=1}^m a_{i,j}$: la projection verticale de la colonne j , $1 \leq j \leq n$.
- $V = (v_1, v_2, \dots, v_n)$: la projection verticale de la matrice A .
- $d_k = \sum_{i+j=k+1} a_{i,j}$: la projection diagonale de rang k , $1 \leq k \leq m + n - 1$.
- $D = (d_1, d_2, \dots, d_{m+n-1})$: la projection diagonale de la matrice A .
- $a_k = \sum_{i+(n-j)=k} a_{i,j}$: la projection anti-diagonale de rang k , $1 \leq k \leq m + n - 1$.
- $A = (a_1, a_2, \dots, a_{m+n-1})$: la projection anti-diagonale de la matrice A .
- DO : un domino est un couple de deux cellules adjacentes. Un domino horizontal (resp. domino vertical), DO_H (resp. DO_V), est un ensemble de deux cellules adjacentes sur la même ligne (resp. colonne).
- $X = (x_1, x_2, \dots, x_t)$, un vecteur de dimension t . X est décroissant si : $x_1 \geq x_2 \geq \dots \geq x_t$ (resp croissant si : $x_1 \leq x_2 \leq \dots \leq x_t$), voir figure 2.1.

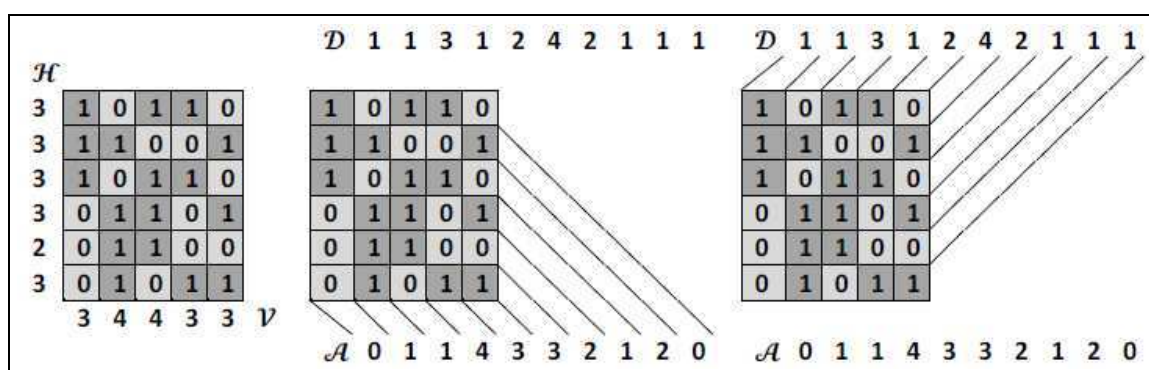


FIGURE 2.1 – Une matrice binaire avec ses projections horizontales H , verticales V , diagonales D et anti-diagonale A

2. Problème de tomographie discrète

Définition 2.2.1 (*adjacence*) Dans une matrice binaire A , chaque élément a 8 voisins : 4 selon les axes horizontaux et verticaux $\{a_{i\pm 1,j}, a_{i,j\pm 1}\}$ et 4 selon la diagonale et l'anti-diagonale $\{a_{i\pm 1,j\pm 1}\}$. On note par q -adjacence, $q \in \{2, 4, 6, 8\}$ la propriété d'adjacence entre $a_{i,j}$ et ses voisins (voir figure 2.2).

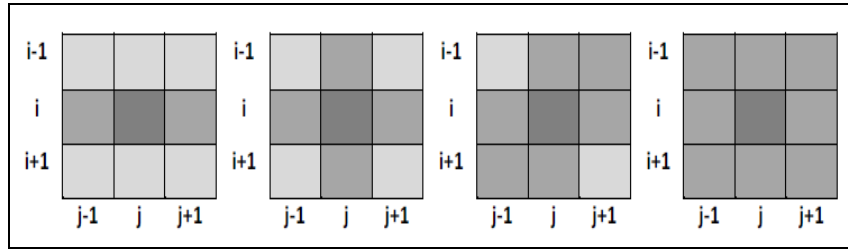


FIGURE 2.2 – Définition de q -adjacence : de gauche à droite $q \in \{2, 4, 6, 8\}$

Définition 2.2.2 (*chemin k -connexe*) Un chemin k -connexe est une suite de points discrets $\{p_i\}_{i=0,\dots,n}$ tel que pour tout i , $p_i (i > 0)$ est k -adjacent à p_{i-1} (voir figure 2.3).

Définition 2.2.3 (*k -connexité*) Une matrice binaire est k -connexe si tous les couples de 1s sont reliés par un chemin k -connexe (voir figure 2.3).

Définition 2.2.4 (*composante k -connexe*) Une composante k -connexe est un ensemble maximal de 1s k -connexe (voir figure 2.3).

Définition 2.2.5 (*polyomino*) Un polyomino est un ensemble maximal de 1s 4-connexe (voir figure 2.3).

2.2.2 Reconstruction d'une matrice binaire

Il est couramment admis que le premier problème de tomographie discrète, étudié par H. Ryser [42] à la fin des années 50 est la reconstruction d'une matrice binaire à partir de ses projections orthogonales (H, V) . Ce problème, noté $BM(H, V)$, consiste à trouver une matrice binaire de projection horizontale H et de projection verticale V . On définit trois problématiques associées à ce problème :

2. Problème de tomographie discrète

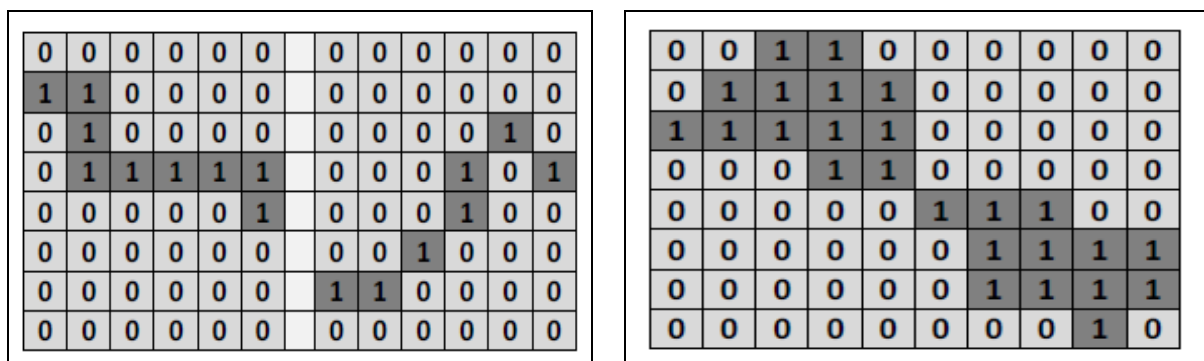


FIGURE 2.3 – De gauche à droite un chemin 4-connexe, un chemin 8-connexe et une matrice binaire contient deux composantes 4-connexe ou une composante 8-connexe

Existence $BM(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Question : Existe-t-il une matrice binaire, qui respecte H et V ?

Reconstruction $BM(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Une matrice binaire, si elle existe, qui respecte H et V .

Unicité $BM(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Une matrice binaire qui respecte H et V , est-elle unique ?

Ryser [42] et Gale [27] ont montré indépendamment que les problèmes : Existence $BM(H, V)$, Reconstruction $BM(H, V)$ et Unicité $BM(H, V)$ sont polynomiaux. Ces questions ont été résolues par le théorème 2.2.1 (Gale et Ryser) et plusieurs algorithmes basés sur ce théorème permettent de reconstruire une matrice binaire [32]. On peut citer, un algorithme glouton qui à chaque colonne j place v_j '1' sur les lignes qui ont les valeurs les plus grandes de h_i .

Théorème 2.2.1 (Gale [27] Ryser [42]) *Existence $BM(H, V)$ a une réponse oui si seulement si :*

$\sum_{i=1}^m h_i = \sum_{j=1}^n v_j$ et $\sum_{j=1}^k \hat{v}_j \geq \sum_{j=1}^k v_j$. Avec $\hat{v}_j = |\{i : h_i \geq j, i = 1, \dots, m\}|$, et $j = 1, \dots, n$.

Définition 2.2.6 (*Opération élémentaire*) Une bascule est un ensemble de quatre cellules (i, j) , $(i, j + k)$, $(i + h, j)$ et $(i + h, j + k)$ tel que (i, j) et $(i + h, j + k)$ sont de valeurs 1 et $(i + h, j)$ et $(i, j + k)$ sont de valeurs 0 ou inversement. Une opération élémentaire consiste à échanger les 1 et 0. Les projections orthogonales d'une matrice binaire restent inchangées après les opérations élémentaires.

Wang et Zhang [45] ont montré que le nombre de solutions de $BM(H, V)$ est très grand. Pour $H = V = (1, 1, \dots, 1)$ il existe $(n!)$ solutions équivalentes telles qu'il est possible de passer d'une solution à une autre par une suite finie d'opérations élémentaires (théorème 2.2.2).

Théorème 2.2.2 (*Ryser [42]*) Si A et B sont deux solutions différentes de $BM(H, V)$, alors A se transforme en B à l'aide d'une suite finie d'opérations élémentaires.

2.2.3 Applications

Les applications les plus importantes de la tomographie discrète sont :

2.2.3.1 Imagerie médicale

Pour un traitement médical complexe, obtenir des informations sur les cellules normales et sur les autres est très important pour le malade. Ces informations décrivent efficacement la situation du malade et permettent de déterminer la nature du traitement à appliquer. La tomographie discrète permet de reconstruire certaines propriétés du corps humain à partir d'une série d'images radiologiques [33].

2.2.3.2 Microscopie électronique

Le QUANTITEM (QUAntitative ANalysis of The Information from Transmission Electron Microscopy) est une technique de microscopie électronique qui compte directement le nombre d'atomes dans un cristal. Ici, les techniques de tomographie discrète peuvent être appliquées efficacement : les projections représentent le nombre d'atomes dans le cristal sur une direction donnée et la présence ou l'absence d'atomes est représentée par un 1 ou par un 0 [7; 44].

2.2.3.3 Compression de données

Sans traitement préalable, l'espace nécessaire au stockage d'une image numérique composée de m lignes et n colonnes est de l'ordre de $O(m.n)$. Le stockage de projections orthogonales de l'image nécessite un espace moindre, de l'ordre de $O(m+n)$. Les projections de départ sont obtenues à partir de l'image de départ et un algorithme de reconstruction va fournir une image.

2.2.3.4 Géométrie convexe

La définition de ensembles convexes dans \mathbb{Z}^n est une question classique de la géométrie discrète. Il existe plusieurs notions de convexité discrète : la convexité simple selon certaines directions, la convexité totale (la convexité usuelle du continu), etc. Une étude sur la notion de convexité discrète et la reconstruction d'ensembles convexes discrets sont approfondis dans [16; 23].

2.2.3.5 Physique statistique

Cette discipline consiste à déterminer les propriétés microscopiques de la matière condensée (énergie, densité, entropie, etc). Dans les modèles des gaz durs [8], deux sites (positions) voisins, ne sont pas occupés simultanément par des particules parce que l'énergie de répulsion entre elles est assez forte. On parlera du modèle des gaz durs carrés (Hard Square Gas) lorsque les particules sont disposées sur un réseau bidimensionnel carré [39]. Toutes les propriétés statistiques des gaz sont déduites de l'énumération de toutes les configurations possibles des particules. Ceci impose d'énumérer les matrices binaires n'ayant pas deux cellules voisines de valeur 1 (voir [34]).

2.2.3.6 Planification de personnel

Le problème de planification de personnel consiste à planifier les horaires de travail de personnel pour couvrir une demande (charge) tout en respectant un ensemble de règles plus au moins rigides (contraintes). C'est un problème pratique très important pour les organismes opérant sept jours par semaine ou 24 heures par jour car il a un impact direct sur le coût de la main d'oeuvre, la qualité du

service, la productivité et la satisfaction des employés (voir [34]).

2.2.3.7 Radio Tomographic Imaging (RTI)

RTI est une technologie qui localise les objets en mouvement dans une zone entourée par des capteurs. Chaque capteur est capable de transmettre et recevoir des signaux sans fil. Ces signaux créent un réseau de signal. Les objets qui se déplacent dans la zone de capteurs empêchent certains signaux d'atteindre leurs destinations. Une image de l'endroit où le signal est absorbé peut être reconstruite en utilisant les mesures de perte de signal. RTI est utile en cas d'urgence, les opérations de sauvetage et les failles de sécurité pour localiser rapidement les objets dans la zone située. Par exemple, RTI pourrait aider les pompiers à sauver des vies en localisant les victimes rapidement [46; 48].

2.3 Méthodes de reconstruction

Plusieurs approches et méthodes sont utilisées pour aborder un problème de reconstruction en tomographie discrète.

2.3.1 Théorie des graphes

Certains problèmes de la tomographie discrète peuvent être reformulés en tant que problèmes de théorie des graphes [6; 20], d'autres correspondent à des problèmes de flot (voir figure 2.4).

2.3.2 Programmation mathématique

Le problème de reconstruction d'un objet, en particulier d'une matrice, peut être formulé comme un système d'équations linéaires $AX = B$. La matrice A décrit la correspondance entre les éléments de l'objet à reconstruire et ses projections, le vecteur de nombre entiers B définit les projections mesurées et X représente les éléments de l'objet à reconstruire.

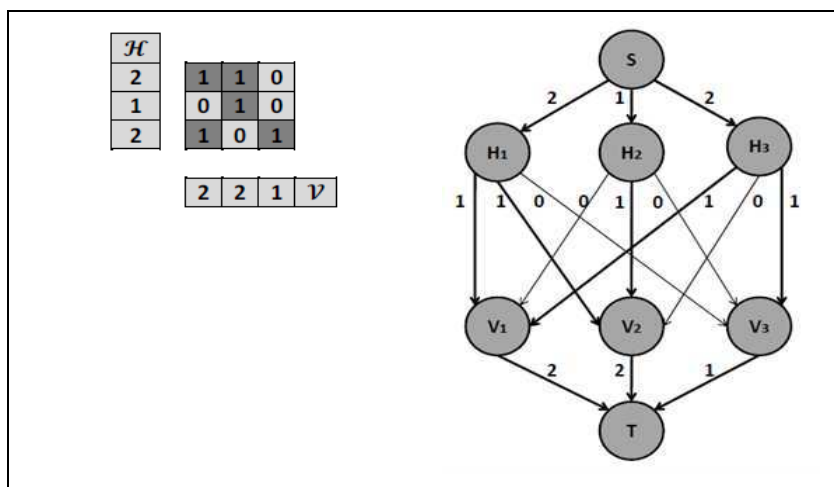


FIGURE 2.4 – Matrice binaire avec ses projections orthogonales et le graphe orienté correspondant

2.3.3 Techniques d'optimisation combinatoire

Actuellement, deux principales familles de méthodes sont connues pour résoudre exactement un programme linéaire en nombres entiers : les méthodes de recherche arborescente (énumération partielle) et les méthodes de coupes (troncatures).

La méthode de recherche arborescente par séparation et évaluation est une méthode générique de résolution de problèmes d'optimisation combinatoire. Elle énumère de manière intelligente l'ensemble des solutions. Pour cela, elle décompose l'espace des solutions en sous-ensembles de plus en plus petits, dont une bonne partie est éliminée à l'aide de bornes. Ce type d'énumération peut donc fournir une solution optimale en un temps réduit par rapport à une énumération complète. Cependant, pour les instances de grande taille des problèmes NP-difficiles, leur durée d'exécution est encore trop importante pour pouvoir être utilisable dans des applications réelles, il faut alors se tourner vers les approches heuristiques. Les composantes principales des méthodes arborescentes sont :

- une règle de séparation des solutions.
- une fonction d'évaluation des solutions.
- une stratégie d'exploration.

Les méthodes de coupes sont basées sur l'idée suivante. On commence par ré-

soudre le programme linéaire continu. Si la solution optimale continue est à coordonnées entières alors l'optimum entier est obtenu et le problème est résolu. Dans le cas contraire (la situation fréquemment rencontrée) on peut toujours tronquer le domaine des solutions (en rajoutant une ou plusieurs contraintes supplémentaires) de façon à éliminer cette solution sans éliminer aucune solution entière. Une telle contrainte est appelée une coupe, ou inégalité valide. Après avoir rajouté une coupe (ou plusieurs) le raisonnement précédent peut être répété avec le programme linéaire augmenté des contraintes correspondantes. Malheureusement, et c'est la difficulté majeure, on ne connaît pas de méthode systématique efficace pour engendrer toutes les inégalités valides (Gomory 1958 [30]).

2.4 Problèmes traités en tomographie discrète

Vu le très grand nombre des solutions du problème $BM(H, V)$ et la difficulté pour contrôler ce nombre exponentiel, il est important d'introduire des techniques pour réduire l'espace des solutions de $BM(H, V)$. En général, il existe deux méthodes ; la première méthode consiste à augmenter le nombre de projections. La deuxième méthode consiste à ajouter des propriétés géométriques sur l'objet (la matrice) à reconstruire (connexité, convexité, périodicité, etc).

Plusieurs difficultés apparaissent dans les deux approches. Par exemple pour la première méthode : si le nombre de projections atteint trois, alors le problème de reconstruction d'une matrice binaire devient NP-complet [24]. Pour la deuxième méthode il est possible pour quelques contraintes géométriques, de trouver des algorithmes polynomiaux mais pour d'autres propriétés le problème de reconstruction demeure NP-complet.

2.4.1 Reconstruction de matrices convexes

La définition géométrique de la convexité dans l'espace euclidien \mathbb{R}^n est très simple : un ensemble $X \subset \mathbb{R}^n$ est convexe si tout segment joignant deux points de X est contenu dans X . Cette définition n'est pas immédiatement transposable dans \mathbb{Z}^n . De nombreuses définitions de la convexité discrète en tomographie discrète sont approfondies dans [23].

2. Problème de tomographie discrète

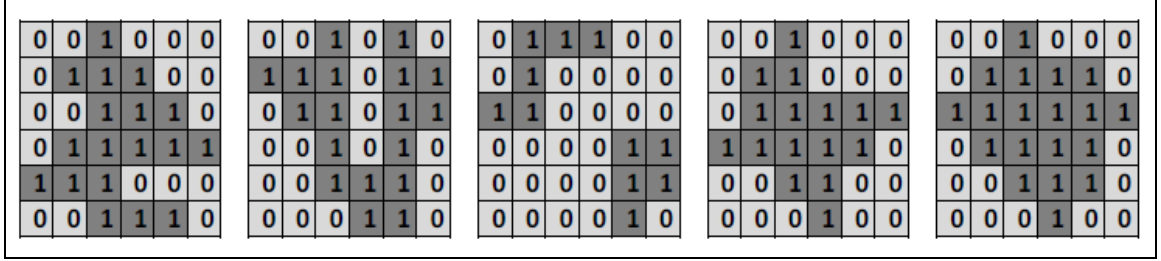


FIGURE 2.5 – De gauche à droite : $\{h, v, hv, hvd, hvda\}$ -convexe

Définition 2.4.1 (*Convexité simple*) Un ensemble A est convexe selon la direction $d = (a, b)$ si pour tous points $X = (i_1, j_1)$ et $Y = (i_2, j_2)$ de A tels que (XY) est parallèle à d , les points entiers du segment $[XY]$ sont dans A . En particulier, une matrice binaire est :

- horizontalement convexe, h -convexe, s'elle est convexe suivant la direction $d = (1, 0)$.
- verticalement convexe, v -convexe, s'elle est convexe suivant la direction $d = (0, 1)$.
- diagonalement convexe, d -convexe, s'elle est convexe suivant la direction $d = (1, 1)$.
- anti-diagonalement convexe, a -convexe, s'elle est convexe suivant la direction $d = (-1, 1)$, (voir figure 2.5).

Définition 2.4.2 (*Convexité par quadrant*) Pour tout point $P = (p_1, p_2)$ on définit les quatre quadrants autour de P par :

- $R_0(P) = \{Q = (q_1, q_2) | q_1 \leq p_1 \text{ et } q_2 \leq p_2\}$
- $R_1(P) = \{Q = (q_1, q_2) | q_1 \geq p_1 \text{ et } q_2 \leq p_2\}$
- $R_2(P) = \{Q = (q_1, q_2) | q_1 \geq p_1 \text{ et } q_2 \geq p_2\}$
- $R_3(P) = \{Q = (q_1, q_2) | q_1 \leq p_1 \text{ et } q_2 \geq p_2\}$

A est Q -convexe (convexe par quadrant) si : $R_k(P) \cap A \neq \emptyset \Rightarrow P \in A, k \in \{0, 1, 2, 3\}$, (voir figure 2.6)

Dans la suite, nous considérons la définition simple de la convexité discrète. On note $T\text{-conv}(H, V)$ le problème de reconstruction d'une matrice binaire vérifiant la propriété de convexité simple (Définition 2.4.1). Les problèmes d'existence et de reconstruction associés sont :

2. Problème de tomographie discrète

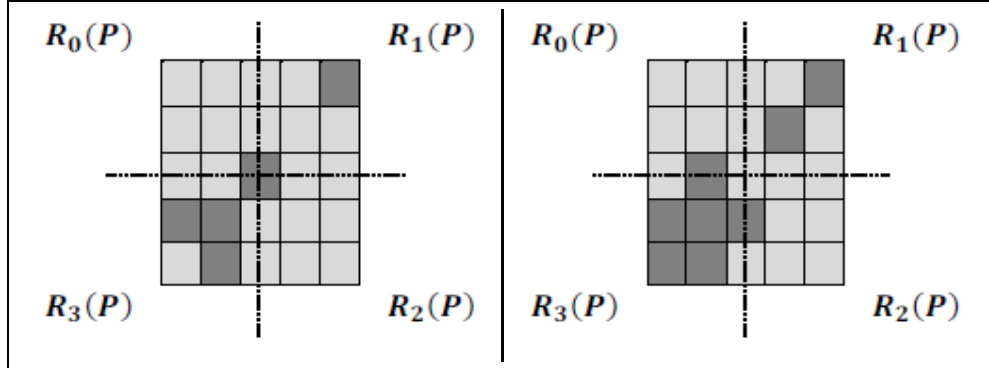


FIGURE 2.6 – A gauche un Q-convexe et à droite un non Q-convexe : $P \notin A$ alors que $R_1(P) \cap A \neq \emptyset$ et $R_4(P) \cap A \neq \emptyset$

Existence $T - conv(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Question : Existe-t-il une matrice binaire $T - convexe$, qui respecte H et V ?

Reconstruction $T - conv(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Une matrice binaire $T - convexe$, si elle existe, qui respecte H et V .

Les résultats de complexité sont résumés dans le tableau 2.1

		Les projections				
		(H, V)	2	(H, V, D)	3	≥ 4
Libre	Libre	P [42]	P [42]	NP-C [28]	NP-C [28]	NP-C [28]
	$\{h, v, d, a\} - convexe$	NP-C [5]	?*	NP-C [16]	?*	?*
	$hw - convexe$	NP-C [47]	?*	?*	?*	?*
	$hvd - convexe$?*	?*	NP-C [16]	?*	?*
Polymino	Libre	NP-C [47]	?*	NP-C [16]	?*	?*
	$\{h, v, d, a\} - convexe$	NP-C [5]	?*	NP-C [16]	?*	?*
	$hw - convexe$	P [5]	?*	?*	?*	?*
	$hvd - convexe$	P [16]	?*	P [16]	?*	?*

TABLE 2.1 – Complexité du problème de reconstruction d'une matrice binaire $T - convexe$. '*': problème ouvert

Nous rappelons qu'un polyomino est un ensemble maximal de 1s 4-connexe.

2.4.1.1 Reconstruction d'une matrice binaire hv-convexe

La classe hv-convexe est l'une des classes la plus importante et la plus étudiée en tomographie discrète (voir [16; 23; 43]). En général, le problème de reconstruction à partir de deux projections dans la classe hv-convexe est NP-complet. Des méthodes d'approximations pour résoudre le problème de reconstruction d'un hv-convexe ont été proposées en appliquant certaines heuristiques, métaheuristiques ou techniques d'optimisation.

La reconstruction d'une matrice binaire convexe à partir de deux projections est NP-complet [5]. La reconstruction d'une matrice binaire h-convexe (resp. v-convexe) est NP-complet. De même, le problème de reconstruction d'un polyomino h-convexe (resp. polyomino v-convexe) est NP-complet. Woeginger [47] a montré aussi que la reconstruction d'un hv-convexe reste NP-complet. Mais la reconstruction d'un hv-convexe et connexe, appelé polyomino hv-convexe, a été montrée polynomiale [5; 18].

La reconstruction une matrice binaire hv-convexe à partir de quatre projections $\{H, V, D, A\}$ reste non déterminée. Peter Balázs [1] a donné une réduction à partir de quatre projections $\{H, V, D, A\}$ pour résoudre un ensemble de sous-problèmes de reconstruction de polyominos hv-convexe à partir de H' et V' tel que $H' \subset H$ et $V' \subset V$ pour chaque sous problème. L'algorithme est constitué de deux étapes. Dans la première étape, l'algorithme construit, à l'aide d'une décomposition de D ou A , les projections H' et V' . Dans la deuxième étape, chaque instance de la reconstruction de polyomino hv-convexe à partir de H' et V' est résolue à l'aide de l'algorithme de Barucci et al. [5].

2.4.2 Approximations et reconstruction d'un hv-convexe

Plusieurs approches d'approximations ont été proposées dans la littérature. On pourrait citer les travaux de : Boufkhad et al. [14], Dahl et Flatberg [22], Costa et al. [35], Jarray et Tlig [36].

L'approche de Dahl et Flatberg [22], consiste à résoudre la relaxation lagrangienne. La décomposition de la relaxation donne $(m + n)$ sous-problèmes poly-

nomiaux dont chacun est équivalent au problème du plus long chemin dans un graphe orienté sans cycle. Dans la suite, nous détaillons la méthode de Dahl et Faltberg.

L'approche Costa et al. [35], consiste à résoudre deux sous problèmes polynomiaux : reconstruire une matrice binaire h-convexe (resp. v-convexe) qui respecte la projection horizontale (resp. verticale). La résolution de chaque sous-problème est basée sur la recherche du plus long chemin dans un graphe orienté sans cycle.

2.4.2.1 Relaxation lagrangienne pour la reconstruction de matrice binaire hv-convexe

Cette méthode a été proposée par Dahl et Faltberg[22]. Le modèle consiste à maximiser le nombre des 1s adjacents dans chaque rangée tout en respectant les projections orthogonales.

Variables de décision Chaque élément de A est associé à une variable binaire x_{ij} , $x_{ij} = a_{ij}$. Le programme est le suivant :

$$P \left\{ \begin{array}{l} \max f(x) = \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j}x_{i+1,j} + \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j}x_{i,j+1} \\ s.c. \\ \sum_{j=1}^n x_{ij} = h_i \quad i = 1, \dots, m \\ \sum_{i=1}^m x_{ij} = v_j \quad j = 1, \dots, n \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{array} \right. \quad \begin{array}{l} (1) \\ (2) \end{array}$$

La contrainte (1) (resp. 2) exprime la satisfaction de la projection horizontale (resp. verticale). La fonction objectif maximise le nombre des '1' adjacents dans les lignes et les colonnes.

Soit le graphe $G = (V, E)$ tel que les sommets V sont les éléments de A et les arêtes E relient les $a_{i,j}$ entre eux sur les lignes et les colonnes. Alors le programme P s'écrit comme le problème de maximisation suivant :

$$P' \left\{ \begin{array}{l} \max f(x) = \sum_{e \in E} y_e \\ \text{s.c.} \\ \sum_{j=1}^n x_{ij} = h_i \quad i = 1, \dots, m \quad (1) \\ \sum_{i=1}^m x_{ij} = v_j \quad j = 1, \dots, n \quad (2) \\ y_e \leq x_{ij} \quad e \in E \quad \text{et} \quad (i, j) \in e \quad (3) \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \\ y_e \in \{0, 1\} \quad e \in E \end{array} \right.$$

Décomposition de variables La technique de décomposition des variables est utilisée par les auteurs pour résoudre la relaxation lagrangienne de P' . Après la duplication de $x_{i,j}$ en $x_{i,j}^h$ et $x_{i,j}^v$ (h pour horizontale et v pour verticale) et l'ajout de la contrainte $x_{i,j}^h = x_{i,j}^v$ le programme P' devient :

$$P' \left\{ \begin{array}{l} \max f(x) = \sum_{e \in E} y_e \\ \text{s.c.} \\ \sum_{j=1}^n x_{ij}^h = h_i \quad i = 1, \dots, m \quad (1) \\ \sum_{i=1}^m x_{ij}^v = v_j \quad j = 1, \dots, n \quad (2) \\ y_e \leq x_{ij}^h \quad e \in E \quad \text{et} \quad (i, j) \in e \quad (3) \\ y_e \leq x_{ij}^v \quad e \in E \quad \text{et} \quad (i, j) \in e \quad (4) \\ x_{ij}^h = x_{ij}^v \quad i = 1, \dots, m; j = 1, \dots, n \quad (5) \\ x_{ij}^h, x_{ij}^v \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \\ y_e \in \{0, 1\} \quad e \in E \end{array} \right.$$

La relaxation de la contrainte $x_{i,j}^h = x_{i,j}^v$ en utilisant les multiplicateurs de Lagrange $\lambda_{i,j}$ donne :

2. Problème de tomographie discrète

$$PRL \left\{ \begin{array}{l}
 \max f(x) = \sum_{e \in E} y_e + \sum_{i,j} \lambda_{i,j} (x_{i,j}^h - x_{i,j}^v) \\
 \text{s.c.} \\
 \sum_{j=1}^n x_{ij}^h = h_i \quad i = 1, \dots, m \quad (1) \\
 \sum_{i=1}^m x_{ij}^v = v_j \quad j = 1, \dots, n \quad (2) \\
 y_e \leq x_{ij}^h \quad e \in E \quad \text{et} \quad (i, j) \in e \quad (3) \\
 y_e \leq x_{ij}^v \quad e \in E \quad \text{et} \quad (i, j) \in e \quad (4) \\
 x_{ij}^h, x_{ij}^v \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \\
 y_e \in \{0, 1\} \quad e \in E
 \end{array} \right.$$

Le programme PRL peut être décomposé en $(m + n)$ sous-problèmes polynomiaux, un sous-problème pour chaque rangée (ligne PRL_i et colonne PRL_j).

$$PRL_i \left\{ \begin{array}{l}
 \text{s.c.} \\
 \sum_{j=1}^n x_{ij}^h = h_i \quad (1) \\
 y_k \leq x_{ij}^h \quad k \in E \quad (3) \\
 x_{ij}^h \in \{0, 1\} \quad j = 1, \dots, n \\
 y_k \in \{0, 1\} \quad k \in E
 \end{array} \right.$$

$$PRL_j \left\{ \begin{array}{l}
 \text{s.c.} \\
 \sum_{i=1}^m x_{ij}^v = v_j \quad (2) \\
 y_k \leq x_{ij}^v \quad k \in E \quad (4) \\
 x_{ij}^v \in \{0, 1\} \quad i = 1, \dots, m \\
 y_e \in \{0, 1\} \quad e \in E
 \end{array} \right.$$

Chaque sous-problème PRL_i ou PRL_j est équivalent au problème de plus long chemin entre deux sommets dans un graphe orienté sans cycle. Une formulation générale pour les sous problèmes PRL_i et PRL_j est la suivante :

2. Problème de tomographie discrète

$$\text{PRL} \begin{cases} \max (\sum_{k=1}^{n-1} y_k + \sum_{k=1}^n c_k x_k) \\ \text{s.c.} \\ \sum_{k=1}^N x_k = b \\ y_k \leq x_k \quad k = 1, \dots, N \\ y_k \leq x_{k+1} \quad k = 1, \dots, N - 1 \\ y_k, x_k \in \{0, 1\} \quad k = 1, \dots, N \end{cases}$$

Tel que N vaut n (resp. m) pour chaque PRL_i (resp. PRL_j), b est h_i (resp. v_j) et c_k le multiplicateur de Lagrange (avec un signe (-) pour les PRL_j). Le problème PRL correspond au problème de plus long chemin dans un graphe orienté sans cycle de $b + 1$ arcs, $G = (V, E)$, dont les sommets V sont les x_k , une source s et un puits t . Les arcs E de G sont étiquetés par les coûts suivants :

- (s, x_k) de coût c_k ,
- (x_{k-1}, x_k) de coût $c_k + 1$,
- (x_l, x_k) pour $l < k - 1$ de coût c_k ,
- $x_{k,t}$ de coût 0.

Chaque instance de PRL est résolue efficacement à l'aide d'un algorithme de programmation dynamique, tel que le plus long chemin entre les sommets (s) et (i) contenant (j) arcs, noté $d(i, j)$, est calculé par la formule de récurrence suivante :

1. $d(i, 1) = c_i$
2. $d(i, j) = \max\{d(u, j - 1) + c_j, \text{ pour } u < i - 1; d(i - 1, j - 1) + c_j + 1\}$
3. La solution finale est $d(t, b + 1)$

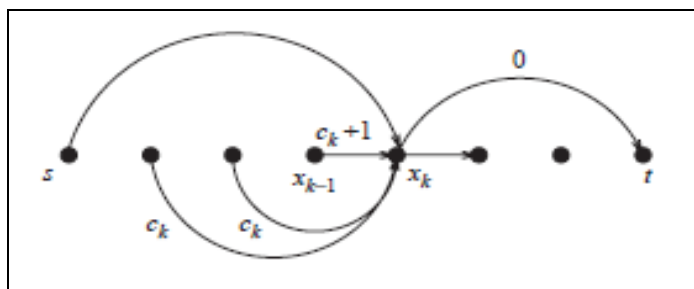


FIGURE 2.7 – La construction d'un graphe pour PRL

2.4.2.2 Reconstruction d'un polyomino hv-convexe

Reconstruire une matrice binaire hv-convexe et connexe peut être résolu en temps polynomial, comme peut l'être aussi le problème de reconstruction d'une matrice binaire sans conditions sur la solution. Essentiellement deux algorithmes ont été proposés pour résoudre ce problème.

Barcucci et al [5], ont donné un algorithme polynomial en deux étapes pour reconstruire un polyomino hv-convexe. L'algorithme est la version discrète d'un algorithme de la tomographie continue. Dans la première étape, l'algorithme construit itérativement deux ensembles de cellules, une enveloppe S et un noyau K , avec la propriété que toutes les solutions M respectent $K \subseteq M \subseteq S$. Dans la deuxième étape et quand il n'est plus possible de réduire S ou d'augmenter K une formule 2-SAT est générée où les variables représentent les cellules $S - K$. Ces deux parties de l'algorithme donnent un temps de calcul en $o(m^4n^4)$.

L'algorithme de Chrobak et Dürr, en 1999 [18], résout le problème directement via une formule logique dont chaque variable booléenne est associée à un élément de la matrice à reconstruire. L'algorithme est polynomial en $o(m.n \min(m^2, n^2))$ et linéaire lorsque $h_i = n$ (resp. $v_j = m$).

2.4.3 Reconstruction de matrices périodiques

Définition 2.4.3 (*matrice périodique*) Une matrice binaire A est dite (p, q) – périodique si $a_{i,j} = a_{i+p,j+q}$, pour $i = 1, \dots, m - p$ et $j = 1, \dots, n - q$.

Del Lungo et al. [38] ont montré que la reconstruction d'une matrice binaire (p, q) – périodique est polynomiale lorsque $(p, q) \in \{(1, 1); (1, q); (p, 1)\}$. Elle est de complexité non déterminée, à notre connaissance, dans le cas général.

Définition 2.4.4 (*matrice alternée périodique*) Une matrice binaire A est dite (p, q) – alternée si $a_{i,j} + a_{i+p,j+q} = 1$, pour $i = 1, \dots, m - p$ et $j = 1, \dots, n - q$.

Jarray [34] a montré que la reconstruction d'une matrice binaire (p, q) – alternée est polynomiale lorsque $(p, q) \in \{(1, 1); (1, q); (p, 1)\}$. Elle est de complexité non déterminée, à notre connaissance, dans le cas général.

2. Problème de tomographie discrète

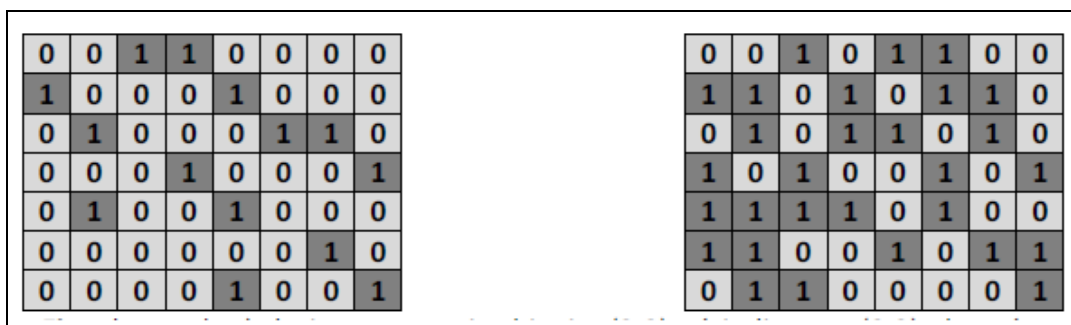


FIGURE 2.8 – De gauche à droite une matrice binaire (2,3)-périodique et (2,3)-alternée

	(1,1)	(p,1) ou (1,q)	(p,q)
(p,q)-périodique	Polynomial [38]	Polynomial [38]	?
(p,q)-alternée	Polynomial [34]	Polynomial [34]	?

TABLE 2.2 – Résultats de complexité du problème de reconstruction d’une matrice périodique.

2.4.4 Reconstruction de matrices adjacentes

La contrainte d’adjacence a été étudiée pour la première fois en 2005 par Costa et al. [21]. Une deuxième variété de cette contrainte a été étudiée dans [15; 17; 26].

Définition 2.4.5 (*Contrainte d’adjacence*) Une matrice binaire respecte la contrainte $p-q$ -adjacence, si $a_{i,j} = 1$, alors $\sum_{a_{t,s} \in \Gamma^q a_{i,j}} a_{t,s} = p$, pour toute $a_{i,j}$. Avec $\Gamma^q a_{i,j}$ les q voisins de $a_{i,j}$. (voir Figure 2.9)

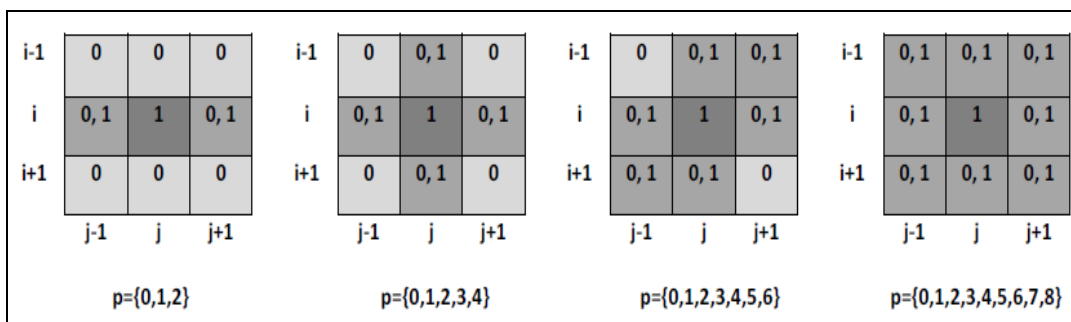


FIGURE 2.9 – Contrainte d’adjacence $p-q$ -adjacence : de gauche à droite $q = \{2, 4, 6, 8\}$

2. Problème de tomographie discrète

La problématique sous la contrainte $p-q$ -adjacence, notée $p-q-adj(H, V)$, est la suivante :

Existence $p-q-adj(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Question : Existe-t-il une matrice binaire qui vérifie la $p-q$ -adjacence, qui respecte H et V ?

Reconstruction $p-q-adj(H, V)$

Données : $H = (h_1, h_2, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Une matrice binaire qui vérifie la $p-q$ -adjacence, si elle existe, qui respecte H et V .

	q=2	q=4	q=6	q=8
p=0	P [21]	?	NP-C [21]	NP-C [21]
p=1	P [41]	NP-C [26]	?	?
p=2	P (trivial)	NP-C [15]	NP-C [15]	NP-C [15]
p ≥ 3	n'est pas défini	?	?	?

TABLE 2.3 – Résultats de complexité du problème $p-q-adj(H, V)$.

Costa et al. [21] ont étudié le problème de reconstruction d'une matrice binaire à partir de H et V sous la contrainte $0-q$ -adjacence pour $q \in \{2, 4, 6, 8\}$. Ils ont montré que le problème est NP-complet pour $q = \{6, 8\}$, de complexité non déterminée pour $q = 4$ et polynomial pour $q = 2$.

En général, la complexité du problème de reconstruction d'une matrice binaire sous la contrainte $0-4$ -adjacence est non déterminée. Les auteurs dans [21] et [17], ont montré que le problème est polynomial pour $m \leq 3$, où si une seule projection orthogonale est respectée.

Picouleau et al. [26] ont montré que le problème de reconstruction d'une matrice binaire à partir de deux projections horizontale H et verticale V sous la contrainte $1-4$ -adjacence est NP-complet. Ils ont montré aussi que la reconstruction sous la contrainte $2-q$ -adjacence est NP-complet si $q \in \{4, 6, 8\}$ [15].

2.4.5 Reconstruction de matrices colorées

Une généralisation possible du problème $BM(H, V)$ est la suivante. Il s'agit de colorier les cellules de la matrice, à raison d'une seule couleur par cellule, en respectant les contraintes données sur le nombre de couleurs dans chaque ligne et chaque colonne. Ce problème sera noté $C - CM(H^C, V^C)$. Les projections sont des vecteurs positifs, $H^c = (h_1^c, \dots, h_m^c)$ pour les lignes et $V^c = (v_1^c, \dots, v_n^c)$, pour les colonnes pour chaque couleur c . La problématique est alors la suivante :

Existence $C - CM(H^C, V^C)$

Données : $H^r = (h_1^r, \dots, h_m^r) \in \mathbb{N}^m$ et $V^r = (v_1^r, \dots, v_n^r) \in \mathbb{N}^n$ pour $r = 1, \dots, C$.

Question : Existe-t-il une matrice C colorée, qui respecte H^r et V^r pour $r = 1, \dots, C$?

Reconstruction $C - CM(H^C, V^C)$

Données : $H^r = (h_1^r, \dots, h_m^r) \in \mathbb{N}^m$ et $V^r = (v_1^r, \dots, v_n^r) \in \mathbb{N}^n$ pour $r = 1, \dots, C$.

Sortie : Une matrice C colorée, si elle existe, qui respecte H^r et V^r pour $r = 1, \dots, C$.

Gardner et al. [29] ont montré que le problème $C - CM(H^C, V^C)$ est NP-complet pour $C \geq 7$. Chobark et Dürr [19] ont démontré que le problème $C - CM(H^C, V^C)$ reste NP-complet pour $C \geq 4$. Récemment Dürr et al. [24] ont montré que le problème $2 - CM(H^2, V^2)$ est aussi NP-complet.

2.5 Conclusion

Cette introduction nous montre qu'il y a peu de résultats dans le domaine de la tomographie discrète. Il est intéressant de donner une description plus précise du problème de reconstruction en se basant sur des propositions et des formulations mathématiques. Puis, étudier et développer de nouvelles approches, en particulier les approches d'optimisation combinatoire et étudier leur efficacité.

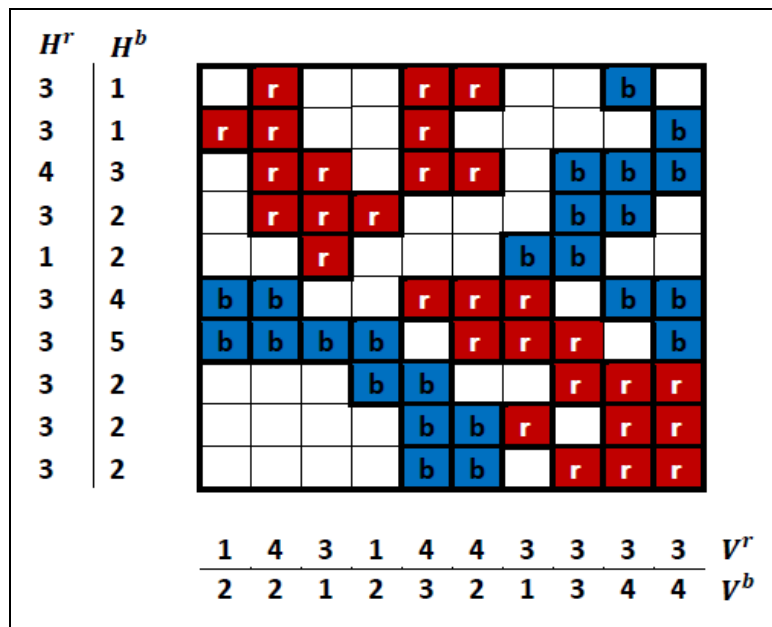


FIGURE 2.10 – Matrice 2-colorée avec ses projections horizontales et verticales pour chaque couleur.

2. Problème de tomographie discrète

Chapitre 3

Programmation mathématique pour la reconstruction des images

Sommaire

3.1	Introduction	27
3.2	Reconstruction d'une image binaire convexe	27
3.3	Programmation quadratique en 0-1 pour reconstruire une image binaire hv-convexe	27
3.3.1	Linéarisation du programme QP	29
3.3.2	Convexification du programme QP	31
3.4	Système algébrique pour reconstruire une image bi- naire convexe	34
3.5	Reconstruction avec variables fixées	35
3.6	Résultats numériques	36
3.6.1	Interprétations	37
3.7	Conclusion	41

Ce chapitre est basé sur les publications suivantes :

1. Reconstructing Convex Matrices by Integer Programming Approaches, *Journal of Mathematical Modelling and Algorithms* 2012 : DOI 10.1007/s10852-012-9193-5
2. Programmation discrète pour la reconstruction de matrices convexes, *ROADEF 2012*. April 2012, Angers, France

3. Programmation mathématique pour la reconstruction des images

3.1 Introduction

Dans ce chapitre, nous considérons le problème de reconstruction d'une image binaire à partir de projections orthogonales. Le problème est défini comme suit : étant donnée une paire de vecteurs entiers positifs H et V , nous cherchons une image binaire convexe consistante avec ces projections H et V .

3.2 Reconstruction d'une image binaire convexe

Nous rappelons la définition de la convexité pour une matrice binaire

Définition 3.2.1 *Une matrice binaire est hv-convexe si les 1s de chaque rangée (ligne et colonne) forment un seul bloc.*

En général, le problème de reconstruction d'une image binaire T – convexe est NP-complet (avec $T \in \{h, v, d, a, hv, hvd, etc\}$ voir Table 2.1). Nous nous intéressons au problème de reconstruction d'une image binaire hv-convexe à partir de projections orthogonales H et V . Nous proposons une formulation basée sur un programme quadratique en 0-1 soumis à des contraintes linéaires.

Nous considérons une image binaire A de taille $m \times n$ et (H, V) deux vecteurs entiers positifs représentant les projections horizontales et verticales de l'image A . Nous rappelons que le problème de reconstruction d'une image binaire hv-convexe, noté $hv - conv(H, V)$, consiste à retrouver à partir de (H, V) une image binaire hv-convexe vérifiant les projections H et V .

$hv - conv(H, V)$

Données : $H = (h_1, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Une image binaire $hv - convexe$ qui respecte H et V .

3.3 Programmation quadratique en 0-1 pour reconstruire une image binaire hv-convexe

Nous introduisons la variable binaire x_{ij} représentant la valeur de l'entrée (i, j) de l'image à reconstruire. La fonction suivante donne le nombre de 1s hori-

3. Programmation mathématique pour la reconstruction des images

zontalement et verticalement adjacents pour une image binaire de taille $m \times n$:

$$f(X) = \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j} x_{i,j+1} + \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j} x_{i+1,j}$$

Nous proposons le programme QP pour reconstruire une image binaire respectant H et V et ayant le nombre maximal de couples de 1s adjacents.

$$QP \begin{cases} \max f(X) \\ \text{s.c.} \\ \sum_{j=1}^n x_{ij} = h_i \quad i = 1, \dots, m & (1) \\ \sum_{i=1}^m x_{ij} = v_j \quad j = 1, \dots, n & (2) \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{cases}$$

Les contraintes (1) et (2) garantissent que l'image à reconstruire satisfait les projections orthogonales. La fonction objectif consiste à maximiser le nombre de 1s adjacents par rangée (ligne et colonne). Nous montrons le résultat suivant :

Proposition 3.3.1 *X est une image binaire hv-convexe respectant les projections H et V si et seulement si $f(X) = 2 \sum_{j=1}^m h_i - m - n(\sigma)$.*

Preuve X est une image binaire hv-convexe si et seulement si les 1s de chaque rangée sont adjacents. Donc

$$\begin{aligned} \sum_{j=1}^{n-1} x_{i,j} x_{i,j+1} &= h_i - 1, \quad i = 1, \dots, m; \quad \sum_{i=1}^{m-1} x_{i,j} x_{i+1,j} = v_{j=1} - 1, \quad j = 1, \dots, n \\ \text{Ainsi } \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j} x_{i,j+1} &= \sum_{i=1}^m h_i - m; \quad \sum_{j=1}^n \sum_{i=1}^{m-1} x_{i,j} x_{i+1,j} = \sum_{j=1}^n v_{j=1} - n \\ \text{D'où } \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j} x_{i,j+1} + \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j} x_{i+1,j} &= \sum_{i=1}^m h_i - m + \sum_{j=1}^n v_{j=1} - n \\ f(X) &= 2 \sum_{j=1}^m h_i - m - n. \end{aligned}$$

■

Le programme QP consiste à maximiser une fonction quadratique non concave sous un ensemble de contraintes linéaires. Le nombre de contraintes (1) et (2) est de l'ordre de $(m + n)$ et le nombre de variables est de l'ordre de $(m \times n)$.

L'optimisation quadratique en variables 0-1 fait l'objet de plusieurs travaux (voir par exemple Boros et Hammer [13], et Billionnet et al. [9; 10; 11; 12]).

3. Programmation mathématique pour la reconstruction des images

En 2006, Billionnet et Elloumi [10] ont étudié l'optimisation d'une fonction quadratique de variables 0-1 sans contraintes. Puis en 2007, Billionnet et al. [12] ont développé une méthode de transformation d'un programme quadratique en variables 0-1 et sous contraintes linéaires en un programme quadratique convexe en variables 0-1. Récemment en 2010, Billionnet et al. [11] ont généralisé cette méthode pour de programmes quadratiques sous contraintes linéaires en nombre entiers.

Afin de faciliter la résolution de QP , nous appliquons des techniques de linéarisation (transformer QP en un programme linéaire) et de convexification (transformer QP en un programme quadratique convexe).

3.3.1 Linéarisation du programme QP

La linéarisation d'un programme quadratique consiste à trouver une reformulation linéaire équivalente. La linéarisation de notre programme QP consiste à remplacer les termes quadratiques de la fonction objectif par des quantités linéaires. Essentiellement, il existe deux méthodes de linéarisation d'un programme quadratique : linéarisation classique et linéarisation compacte.

Linéarisation classique. La linéarisation classique consiste à remplacer chaque produit $x_{i,j} \times x_{i,j+1}$ (resp. $x_{i,j} \times x_{i+1,j}$) par une variable binaire $yh_{i,j}$ (resp. $yv_{i,j}$) et à introduire les inégalités suivantes pour assurer l'égalité $yh_{i,j} = x_{i,j} \times x_{i,j+1}$

$$\begin{cases} yh_{i,j} \leq x_{i,j} & i = 1, \dots, m; j = 1, \dots, n - 1 & (3) \\ yh_{i,j} \leq x_{i,j+1} & i = 1, \dots, m; j = 1, \dots, n - 1 & (4) \end{cases}$$

De même on introduit les inéquations suivantes pour assurer l'égalité $yv_{i,j} = x_{i,j} \times x_{i+1,j}$

$$\begin{cases} yv_{i,j} \leq x_{i,j} & i = 1, \dots, m - 1; j = 1, \dots, n & (5) \\ yv_{i,j} \leq x_{i+1,j} & i = 1, \dots, m - 1; j = 1, \dots, n & (6) \end{cases}$$

3. Programmation mathématique pour la reconstruction des images

On obtient un programme linéaire en variables binaires $LP1$, équivalent à QP , dont la fonction objectif est linéaire :

$$LP1 \left\{ \begin{array}{l} \max f_L(X) = \sum_{i=1}^m \sum_{j=1}^{n-1} y h_{i,j} + \sum_{i=1}^{m-1} \sum_{j=1}^n y v_{i,j} \\ s.c. \\ \sum_{j=1}^n x_{i,j} = h_i \quad i = 1, \dots, m \quad (1) \\ \sum_{i=1}^m x_{i,j} = v_j \quad j = 1, \dots, n \quad (2) \\ y h_{i,j} \leq x_{i,j} \quad i = 1, \dots, m; j = 1, \dots, n-1 \quad (3) \\ y h_{i,j} \leq x_{i,j+1} \quad i = 1, \dots, m; j = 1, \dots, n-1 \quad (4) \\ y v_{i,j} \leq x_{i,j} \quad i = 1, \dots, m-1; j = 1, \dots, n \quad (5) \\ y v_{i,j} \leq x_{i+1,j} \quad i = 1, \dots, m-1; j = 1, \dots, n \quad (6) \\ x_{i,j} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{array} \right.$$

S'agissant d'un problème de maximisation on remarque que le système (I) est équivalent à $2 y h_{i,j} \leq x_{i,j} + x_{i,j+1}$. De même, on remarque que le système (II) est équivalent à $2 y h_{i,j} \leq x_{i,j} + x_{i,j+1}$. Ainsi on obtient un deuxième programme binaire équivalent

$$LP2 \left\{ \begin{array}{l} \max f_L(X) = \sum_{i=1}^m \sum_{j=1}^{n-1} y h_{i,j} + \sum_{i=1}^{m-1} \sum_{j=1}^n y v_{i,j} \\ s.c. \\ \sum_{j=1}^n x_{i,j} = h_i \quad i = 1, \dots, m \quad (1) \\ \sum_{i=1}^m x_{i,j} = v_j \quad j = 1, \dots, n \quad (2) \\ 2 y h_{i,j} \leq x_{i,j} + x_{i,j+1} \quad i = 1, \dots, m; j = 1, \dots, n-1 \quad (3) \\ 2 y v_{i,j} \leq x_{i,j} + x_{i+1,j} \quad i = 1, \dots, m-1; j = 1, \dots, n \quad (4) \\ x_{i,j} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \\ y h_{i,j} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n-1 \\ y v_{i,j} \in \{0, 1\} \quad i = 1, \dots, m-1; j = 1, \dots, n \end{array} \right.$$

Linéarisation compacte. La linéarisation compacte consiste à remplacer le produit entre une variable binaire et une fonction linéaire par une variable entière.

La fonction objectif peut s'écrire sous la forme équivalente suivante :

$$\begin{aligned} f(X) &= \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j} x_{i,j+1} + \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j} x_{i+1,j} \\ &= \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} x_{i,j} (x_{i,j+1} + x_{i+1,j}) + \sum_{j=1}^{n-1} x_{m,j} x_{m,j+1} + \sum_{j=1}^n x_{i,n} x_{i+1,n} \end{aligned}$$

Ainsi, nous remplaçons chaque produit $x_{i,j}(x_{i,j+1} + x_{i+1,j})$ dans $f(X)$ par une

3. Programmation mathématique pour la reconstruction des images

variable $z_{i,j}$ et nous introduisons les contraintes d'équivalence suivantes pour $i = 1, \dots, n$ et $j = 1, \dots, n$:

$$\begin{cases} L_{i,j}x_{i,j} \leq z_{i,j} \leq U_{i,j}x_{i,j} & (7) \\ (x_{i,j+1} + x_{i+1,j}) - U_{i,j}(1 - x_{i,j}) \leq z_{i,j} \leq yv_{i,j} \leq (x_{i,j+1} + x_{i+1,j}) - L_{i,j}(1 - x_{i,j}) & (8) \end{cases}$$

où $L_{i,j}$ et $U_{i,j}$ sont les bornes inférieures et supérieures des fonctions linéaires $(x_{i,j+1} + x_{i+1,j})$. En particulier pour $(x_{i,j+1} + x_{i+1,j})$ on prend $L_{i,j} = 0$ et $U_{i,j} = 2$. On obtient un programme linéaire compact *LCP* équivalent à *QP*.

$$PLCompact \left\{ \begin{array}{l} \max \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} z_{i,j} + \sum_{j=1}^{n-1} yh_j + \sum_{i=1}^{m-1} yv_i \\ \text{s.c.} \\ \sum_{j=1}^n x_{i,j} = h_i \quad i = 1, \dots, m & (1) \\ \sum_{i=1}^m x_{i,j} = v_j \quad j = 1, \dots, n & (2) \\ z_{i,j} \leq 2x_{i,j} \quad i = 1, \dots, m-1; j = 1, \dots, n-1 & (7) \\ z_{i,j} \leq x_{i,j+1} + x_{i+1,j} \quad i = 1, \dots, m-1; j = 1, \dots, n-1 & (8) \\ yh_j \leq x_{m,j} \quad j = 1, \dots, n & (3) \\ yh_j \leq x_{m,j+1} \quad j = 1, \dots, n-1 & (4) \\ yv_i \leq x_{i,n} \quad i = 1, \dots, m & (5) \\ yv_i \leq x_{i+1,n} \quad i = 1, \dots, m-1 & (6) \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{array} \right.$$

3.3.2 Convexification du programme *QP*

La convexification d'un programme quadratique non convexe consiste à trouver une reformulation équivalente quadratique et convexe (resp. concave pour les problèmes de maximisation). Il existe deux familles de convexification : reformulation et relaxation. La méthode de reformulation (perturbation) consiste à modifier la fonction objectif de *QP* pour la rendre convexe. Nous considérons dans ce chapitre plusieurs méthodes de reformulation (perturbations). Le programme *QP* peut s'écrire sous la forme matricielle suivante :

$$QP : \{ \max f(X) = X^T Q X; AX = B, X \in \{0, 1\}^{m \times n} \}$$

3. Programmation mathématique pour la reconstruction des images

Convexification simple La convexification simple est la méthode de perturbation la plus ancienne et la plus simple. Elle consiste à récrire la fonction objectif $f(X)$ en ajoutant le terme $\{-\lambda \sum_{i=1}^m \sum_{j=1}^n (x_{i,j}^2 - x_{i,j})\}$ avec λ est un paramètre réel constant ajouté à chaque terme diagonal de Q tel que la fonction $\{f_{c(\lambda)}(X) = f(X) - \lambda \sum_{i=1}^m \sum_{j=1}^n (x_{i,j}^2 - x_{i,j})\}$ soit concave. Nous rappelons la propriété $x_{i,j}^2 = x_{i,j}$ pour $x_{i,j} \in \{0, 1\}$. Nous obtenons le modèle suivant

$$CP1 : \left\{ \begin{array}{l} \text{Max } f_{c(\lambda)}(X) = X^T Q_\lambda X - \lambda \sum_{i=1}^m \sum_{j=1}^n (x_{i,j}^2 - x_{i,j}); \\ AX = B, X \in \{0, 1\}^{m \times n} \end{array} \right\}$$

Une autre méthode de convexification simple consiste à retrancher la quantité positive $\tau = \lambda \sum_{i=1}^m h_i = \lambda \sum_{i=1}^m i = 1^m \sum_{j=1}^n x_{i,j} = \lambda \sum_{i=1}^m i = 1^m \sum_{j=1}^n x_{i,j}^2$ de $f(X)$. Nous obtenons la forme équivalente suivante :

$$CP2 : \{ \text{Max } f_{c(\lambda)}(X) = X^T Q_\lambda X - \lambda \tau; AX = B, X \in \{0, 1\}^{m \times n} \}$$

Nous allons déterminer la meilleur valeur du paramètre λ

Définition 3.3.1 Soit M une matrice carrée d'ordre n , la plus grande valeur propre de M est appelée le rayon spectral de M et est notée $\rho(M) = \max_{1 \leq i \leq n} (\lambda_i)$, ou (λ_i) sont les valeurs propres de M .

Nous rappelons le théorème suivant.

Théorème 3.3.1 Si $\|\cdot\|$ est une norme matricielle consistante alors $\rho(Q) \leq \|Q\|$

Nous élaborons la proposition suivante.

Proposition 3.3.2 $\lambda \leq 2$

Preuve $\lambda = \rho(Q) \leq \|Q\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n q_{ij}$, avec $\|Q\|_\infty$ la norme infinie de Q .

■

3. Programmation mathématique pour la reconstruction des images

Ici, on prend $\lambda = \rho(Q) = 2$, pour obtenir le programme convexe équivalent à QP

$$CP2 \quad \begin{cases} \max f_{C(2)}(X) = f(X) - 2 \sum_{i=1}^m \sum_{j=1}^n x_{i,j}^2 \\ \text{s.c.} \\ \sum_{j=1}^n x_{i,j} = h_i \quad i = 1, \dots, m & (1) \\ \sum_{i=1}^m x_{i,j} = v_j \quad j = 1, \dots, n & (2) \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{cases}$$

Convexification QCR : La convexification QCR est une méthode de perturbation récente. Elle consiste à perturber tous les termes de la matrice Q de la fonction initiale par l'ajout de fonctions quadratiques nulles sur l'ensemble de solutions en utilisant à la fois les contraintes d'intégrité et les contraintes d'égalité suivantes :

1. $\sum_{i=1}^m \sum_{j=1}^n (x_{i,j}^2 - x_{i,j})$: Les contraintes d'intégrité.
2. $\sum_{i=1}^m (\sum_{j=1}^n \alpha_{i,j} x_{i,j})(\sum_{j=1}^n x_{i,j} - h_i)$: Les contraintes d'égalité pour les projections horizontales.
3. $\sum_{j=1}^n (\sum_{i=1}^m \beta_{i,j} x_{i,j})(\sum_{i=1}^m x_{i,j} - v_j)$: Les contraintes d'égalité pour les projections verticales.

En pratique, la méthode QCR sous sa forme compacte [12] consiste à perturber tous les termes de la matrice Q en utilisant seulement l'expression linéaire de carrées.

$$f_\alpha(X) = \alpha \sum_{i=1}^m \left(\sum_{j=1}^n (x_{i,j} - h_i)^2 \right); g_\alpha(X) = \sum_{j=1}^n \left(\sum_{i=1}^m x_{i,j} - v_j \right)^2; f_\lambda(X) = \sum_{j=1}^m \sum_{j=1}^n \lambda_{i,j} (x_{i,j}^2 - x_{i,j})$$

La nouvelle fonction objectif est : $f_{C(\alpha,\lambda)}(X) = f(X) + f_\alpha(X) + g_\alpha(X) + f_\lambda(X)$.

Par la suite QP s'écrit sous la forme équivalente suivante :

$$QCR : \{ \text{Max } f_{C(\alpha,\lambda)}(X) = f(X) + f_\alpha(X) + g_\alpha(X) + f_\lambda(X); AX = B, X \in \{0, 1\}^{m \times n} \}$$

3. Programmation mathématique pour la reconstruction des images

Ou

$$QCR \quad \begin{cases} \max f_{C(\alpha,\lambda)}(X) = f(X) + f_\alpha(X) + g_\alpha(X) + f_\lambda(X) \\ s.c. \\ \sum_{j=1}^n x_{i,j} = h_i \quad i = 1, \dots, m \\ \sum_{i=1}^m x_{i,j} = v_j \quad j = 1, \dots, n \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{cases} \quad \begin{matrix} (1) \\ (2) \end{matrix}$$

3.4 Système algébrique pour reconstruire une image binaire convexe

Une image binaire est hv-convexe si seulement si il existe une seule barre horizontale (ensemble de 1s adjacents horizontalement) sur chaque ligne et une seule barre verticale (ensemble de 1s adjacents verticalement) sur chaque colonne.

Nous proposons un système algébrique (S) pour résoudre le problème de reconstruction d'une image hv-convexe. Nous introduisons deux variables binaires r_{ij} et c_{ij} avec $r_{ij} = 1$, si une barre horizontale commence au cellule (i, j) et $c_{ij} = 1$, si une barre verticale commence au cellule (i, j) .

$$S \quad \begin{cases} \sum_{j=1}^n x_{i,j} = h_i \quad i = 1, \dots, m & (1) \\ \sum_{i=1}^m x_{i,j} = v_j \quad j = 1, \dots, n & (2) \\ \sum_{j=1}^{n-h_i+1} r_{i,j} = 1 \quad i = 1, \dots, m & (3.a) \\ \sum_{k=j}^{j+h_i-1} x_{i,k} \geq h_i r_{i,j} \quad i = 1, \dots, m; j = 1, \dots, n - h_i + 1 & (3.b) \\ \sum_{k=1}^{j-1} x_{i,k} \leq h_i(1 - r_{i,j}) \quad i = 1, \dots, m; j = 2, \dots, n - h_i & (3.c) \\ \sum_{i=1}^{m-v_j+1} c_{i,j} = 1 \quad j = 1, \dots, n & (4.a) \\ \sum_{k=i}^{i+v_j-1} x_{k,j} \geq v_j c_{i,j} \quad i = 1, \dots, m - v_j + 1; j = 1, \dots, n & (4.b) \\ \sum_{k=1}^{i-1} x_{k,j} \leq v_j(1 - c_{i,j}) \quad i = 2, \dots, m - v_j; j = 1, \dots, n & (4.c) \\ x_{ij}; r_{ij}; c_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n \end{cases}$$

Les contraintes 3.a, 3.b et 3.c (resp. 4.a, 4.b et 4.c) garantissent que si $r_{i,j} = 1$ alors une seule barre horizontale sur la ligne i commence à la colonne j (resp. si $c_{i,j} = 1$ alors une seule barre verticale sur la colonne j commence à la ligne i).

Nous remarquons que les contraintes 3.c et 4.c sont redondantes. La contrainte

3. Programmation mathématique pour la reconstruction des images

3.c peut être obtenue en combinant 1 et 3.b. Considérons la contrainte 1 :

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{i,j} = h_i \Rightarrow \sum_{k=1}^{j-1} x_{i,k} + \sum_{k=j}^{j+h_i-1} x_{i,k} + \sum_{k=j+h_i}^n x_{i,k} = h_i \\ \Rightarrow \sum_{k=1}^{j-1} x_{i,k} + \sum_{k=j}^{j+h_i-1} x_{i,k} \leq h_i \end{array} \right.$$

On applique la contrainte 3.b :

$$\left\{ \begin{array}{l} \sum_{k=j}^{j+h_i-1} x_{i,k} \geq h_i r_{i,j} \Rightarrow \sum_{k=1}^{j-1} x_{i,k} \leq h_i - \sum_{k=j}^{j+h_i-1} x_{i,k} \leq h_i - h_i r_{i,j} \\ \sum_{k=1}^{j-1} x_{i,k} \leq h_i(1 - r_{i,j}) \Rightarrow (3.c) \end{array} \right.$$

De même, on montre que la contrainte 4.c est aussi redondante.

3.5 Reconstruction avec variables fixées

On peut améliorer toutes les formulations en effectuant des opérations de pétraitement pour certaines variables $x_{i,j}$.

Définition 3.5.1 *Une cellule (i, j) est 1-invariante (resp. 0-invariante) si $x_{i,j} = 1$ (resp. $x_{i,j} = 0$) pour tout $X \in BM\{H, V\}$. Une cellule (i, j) est invariante si elle est 0-invariante ou 1-invariante.*

Le théorème suivant donne les conditions nécessaires et suffisantes pour les cellules invariantes.

Théorème 3.5.1 *(Haber [31]) Étant donnés H et V deux vecteurs entiers monotones et $X \in BM(H, V)$. Supposons que la cellule (i, j) est 1-invariante, alors X est de la forme*

$$\begin{bmatrix} S & A1 \\ A2 & W \end{bmatrix}$$

Avec : S est une image de taille $u \times v$, $u \geq i$ et $v \geq j$, où toutes les entrées valent 1.

W est une image de taille $m - u \times n - v$, où toutes les entrées valent 0.

$A1$ et $A2$ sont deux images binaires de taille $u \times n - v$ et $m - u \times v$.

Pour trouver les éléments invariants, il faut calculer une solution initiale $X \in BM(H, V)$ quelconque. Puis, trier les projections et mettre l'image sous la forme

3. Programmation mathématique pour la reconstruction des images

de Haber. Enfin, Les éléments invariants sont les entrées de l'image S et W de Haber[31]. Pour chercher une solution initiale nous utilisons l'algorithme de Ryser suivant [42] :

Algorithme de Ryser

* Données : $H = (h_1, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$.

* Sotie : A , une image binaire, qui respecte H et V .

1. Pour ($j = 1, \dots, n$) faire

1.1. Si (Il existe v_j lignes ayant $h_i > 0$) alors

Placer (v_j) '1' sur les lignes ayant les (h_i) les plus grands.

Pour tout '1' placé sur une ligne i : $h_i \leftarrow h_i - 1$.

1.2. Sinon, il n'existe pas de solutions. Fin

1.3. Fin si

2. Fin

L'algorithme est glouton de complexité $O(mn + \max(m \log m; n \log n))$. A chaque colonne j , il place v_j '1' sur les lignes qui ont les valeurs les plus grandes de h_i .

3.6 Résultats numériques

Nous avons exécuté les programmes sur un PC AMD Athlon XP-M 1.7 GHz avec 512 Mo de RAM et nous avons utilisé le solveur Cplex 11.

Pour tester la performance de techniques de linéarisation et de convexification, nous avons utilisé les instances décrits dans [2; 3; 4]. Dans [2], Balázs présente une méthode systématique pour générer aléatoirement des images binaires hv-convexe. Les algorithmes proposés sont basés sur la technique de décomposition étudiée dans [25].

Les résultats des méthodes de linéarisation et convexification sont présentés dans les tableaux 3.1 et 3.2. Le tableau 3.6 montre la différence entre le programme $LP1$ avec et sans prétraitement, en utilisant le théorème de Haber, et le système algébrique. Dans les trois tables, la première colonne contient la taille de l'image et le nombre de composantes hv-convexes. Par exemple la ligne (10, 3) concerne une image de taille 10×10 avec 3 composantes hv-convexes sachant qu'une composante hv-convexe est un polyomino hv-convexe. La deuxième colonne contient

3. Programmation mathématique pour la reconstruction des images

le nombre maximal de $1s$ adjacents. Les colonnes Sol , \widehat{Sol} et $Temps$ contiennent respectivement la solution entière, la solution continue et le temps de résolution pour chaque programme.

3.6.1 Interprétations

Nous remarquons que les méthodes de linéarisation sont très similaires entre elles et que la solution optimale pour les images de taille 10×10 est atteinte par toutes les méthodes. D'autre part, nous concluons d'une façon générale que les méthodes de linéarisation sont beaucoup plus efficaces que les méthodes de convexification. En particulier le programme $LP1$ est le meilleur par rapport aux programmes linéaires et le programme QCR est le meilleur par rapport aux programmes convexes. Enfin, nous remarquons que le prétraitement appliqué pour $LP1$ a beaucoup accéléré la résolution en fonction du nombre de variables fixées et que le système algébrique est le plus rapide pour les instances de petite taille.

La figure 3.1 illustre un exemple de reconstruction approchée d'une image binaire convexe avec le programme linéaire $LP1$.

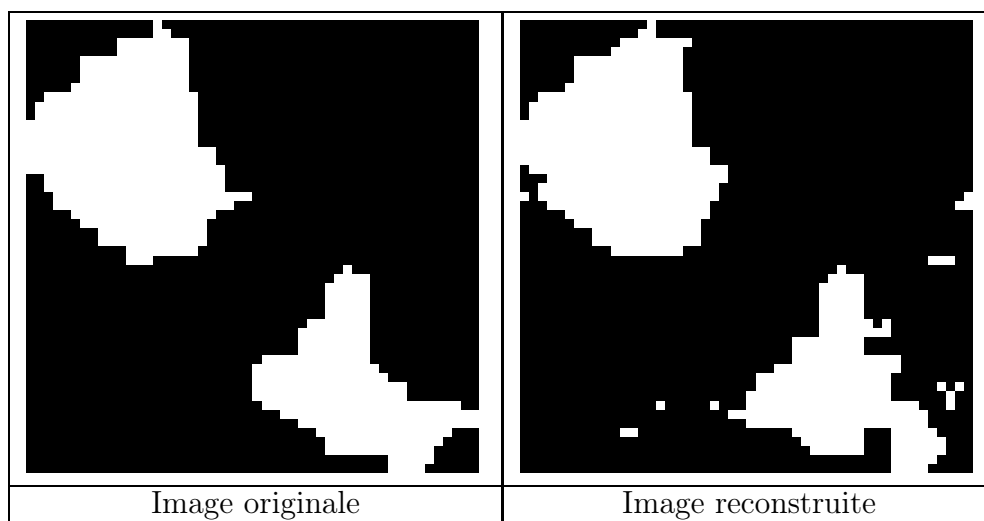


FIGURE 3.1 – Exemple de reconstruction d'une image binaire convexe

#matrix	σ	LP1			LP2			LP3		
		Sol	\widehat{Sol}	Temps	Sol	\widehat{Sol}	Temps	Sol	\widehat{Sol}	Temps
10 × 10,1	80	80	80.88	0.74	80	80.52	1.60	80	80.89	0.78
10 × 10,2	36	36	38.15	1.52	36	37.45	1.59	36	37.44	1.40
10 × 10,3	18	18	20.09	1.34	18	18.67	0.48	18	19.18	1.35
10 × 10,4	20	20	22.07	4.99	20	21.06	1.99	20	21.20	6.80
20 × 20,1	390	390	390.00	0.21	390	390.00	0.26	390	390.00	0.32
20 × 20,2	154	154	155.07	208.64	154	155.15	260.47	154	155.00	1783.56
20 × 20,3	90	90	91.00	1925.23	90	91.06	967.48	85	94.93	3600.18
20 × 20,4	46	46	46.97	213.40	46	46.65	203.23	46	46.84	3024.80
30 × 30,1	954	954	955.00	34.08	954	954.54	78.38	954	955.05	106.08
30 × 30,2	410	410	410.38	165.15	410	410.77	986.73	410	411.02	3368.72
30 × 30,3	240	239	245.42	3600.10	234	245.30	3600.18	234	249.13	3600.10
30 × 30,4	194	193	198.11	3600.10	194	197.09	3600.13	188	200.59	3600.11
40 × 40,1	1390	1366	1398.14	3600.24	1353	1402.21	3600.10	1338	1403.15	3600.10
40 × 40,2	762	757	766.64	3600.11	748	767.59	3600.08	732	770.02	3600.17
40 × 40,3	512	492	520.37	3600.11	488	519.49	3600.02	480	522.22	3600.16
40 × 40,4	274	253	288.00	3600.53	265	283.00	3600.19	258	290.82	3600.13
50 × 50,1	2230	2230	2231.09	424.11	2228	2232.40	3600.17	2228	2231.54	3600.13
50 × 50,2	1166	1133	1172.61	3600.24	1124	1174.17	3600.16	1103	1176.11	3600.22
50 × 50,3	884	882	886.46	3600.16	838	888.76	3600.22	796	891.59	3600.11
50 × 50,4	594	564	605.33	3600.32	570	606.26	3600.11	531	611.04	3601.36

TABLE 3.1 – Résultats de reconstruction par les programmes linéaires

#matrix	σ	<i>CP1</i>			<i>CP2</i>			<i>QCR</i>		
		Sol	\widehat{Sol}	Temps	Sol	\widehat{Sol}	Temps	Sol	\widehat{Sol}	Temps
10 × 10,1	80	80	80.01	99.38	80	80.01	103.42	80	80.12	43.22
10 × 10,2	36	36	38.17	3603.06	36	38.31	3603.29	36	36.99	967.05
10 × 10,3	18	18	18.00	2661.71	18	18.01	2518.47	18	18.01	430.83
10 × 10,4	20	20	24.28	3603.30	20	23.94	3602.75	20	23.64	3600.07
20 × 20,1	390	390	391.00	3.04	390	391.02	2.94	390	391.00	3600.04
20 × 20,2	154	151	182.46	3601.36	148	182.39	3601.94	154	180.02	3600.00
20 × 20,3	90	84	120.12	3600.36	84	120.18	3601.38	90	100.01	3600.10
20 × 20,4	46	42	75.33	3601.72	42	75.37	3601.27	45	75.33	3600.21
30 × 30,1	954	954	978.52	3601.07	954	978.45	3601.00	954	978.40	3600.45
30 × 30,2	410	394	459.13	3600.67	394	459.08	3600.77	400	450.21	3600.17
30 × 30,3	240	216	290.12	3600.70	217	290.08	3600.69	222	280.18	3600.28
30 × 30,4	194	179	243.64	3600.71	172	243.57	3600.64	188	200.99	3600.41
40 × 40,1	1390	1370	1458.08	3601.62	1359	1458.08	3600.36	1385	1410.03	3600.40
40 × 40,2	762	747	832.29	3600.34	738	832.32	3600.33	750	800.32	3600.01
40 × 40,3	512	473	582.08	3600.39	469	582.08	3600.41	482	553.22	3600.00
40 × 40,4	274	238	346.04	3600.58	232	346.03	3600.36	244	346.22	3600.37
50 × 50,1	2236	2208	2295.74	3600.44	2213	2295.80	3600.42	2222	2282.48	3600.05
50 × 50,2	1166	1119	1255.46	3600.19	1118	1255.44	3600.17	1120	1222.77	3600.17
50 × 50,3	884	840	973.81	3600.16	854	973.78	3600.17	850	956.41	3600.02
50 × 50,4	594	533	685.33	3600.27	543	685.33	3600.33	540	665.30	3600.85

TABLE 3.2 – Résultats de reconstruction par les programmes convexes

#martix	σ	<i>LP1</i>			<i>LP1 avec variables fixées</i>				Système <i>S</i>	
		Sol	\widehat{Sol}	Temps	Sol	\widehat{Sol}	Temps	% Fixed	Sol	Temps
10 × 10,1	80	80	80.88	0.74	80	80.00	0.74	0 %	80	0.04
10 × 10,2	36	36	38.15	1.52	36	37.00	1.02	2 %	36	0.09
10 × 10,3	18	18	20.09	1.34	18	18.00	1.03	5 %	18	0.07
10 × 10,4	20	20	22.07	4.99	20	20.02	3.09	7 %	20	0.28
20 × 20,1	390	390	390.00	0.21	390	390.00	0.21	0 %	390	0.06
20 × 20,2	154	154	155.07	208.64	154	155.07	208.64	0 %	154	36.92
20 × 20,3	90	90	91.00	1925.23	90	91.00	1925.23	0 %	85	64.74
20 × 20,4	46	46	46.97	213.40	46	46.00	150.40	10 %	46	24.72
30 × 30,1	954	954	955.00	34.08	954	955.00	3600.19	0 %	954	0.79
30 × 30,2	410	410	410.38	165.15	410	410.00	3600.17	7 %	410	9.32
30 × 30,3	240	239	245.42	3600.10	240	240.13	3600.22	3 %	240	33.29
30 × 30,4	194	193	198.11	3600.10	194	195.51	3600.00	5 %	194	584.13
40 × 40,1	1390	1366	1398.14	3600.24	1390	1390.00	3600.50	0 %	1390	3015.45
40 × 40,2	762	757	766.64	3600.11	762	762.62	3600.01	9 %	762	3600.07
40 × 40,3	512	492	520.37	3600.11	510	514.33	3600.30	10 %	-	3600.00
40 × 40,4	274	253	288.00	3600.53	270	280.00	3600.12	15 %	-	3600.00
50 × 50,1	2230	2230	2231.09	424.11	2230	2230.03	202.10	20 %	2230	133.42
50 × 50,2	1166	1133	1172.61	3600.24	1155	1169.54	3600.22	13 %	-	3600.00
50 × 50,3	884	882	886.46	3600.16	884	885.76	3600.31	4 %	-	3600.00
50 × 50,4	594	564	605.33	3600.32	589	598.81	3600.02	31 %	-	3600.00

TABLE 3.3 – Résultats de reconstruction par *LP1* et le système *S* avec variables fixées ; (-) : solution non trouvée au bout d'une heure de calcul

3.7 Conclusion

Dans ce chapitre, nous avons considéré le problème de la reconstruction des images binaires convexes. Nous avons formulé le problème comme un programme quadratique en nombres entiers, et nous avons proposé des techniques basées sur la linéarisation et la convexification pour la résolution. Nous envisageons prochainement d'aborder les approches polyédrales pour résoudre le problème de la reconstruction et améliorer la qualité des solutions obtenues.

3. Programmation mathématique pour la reconstruction des images

Chapitre 4

Résolution de cas polynomiaux

Sommaire

4.1	Introduction	43
4.2	Reconstruction polynomiale d'une image binaire convexe	44
4.2.1	Reconstruction à partir de deux projections constantes	44
4.2.2	Reconstruction à partir d'une seule projection constante	47
4.3	Reconstruction d'une image binaire h-convexe à partir d'une projection verticale	50
4.4	Reconstruction d'une image binaire hv-convexe avec projection bornée	52
4.5	Conclusion	53

4.1 Introduction

En tomographie discrète, il n'existe pas un algorithme polynomial (sauf si $P = NP$) pour reconstruire une image binaire hv-convexe à partir de projections horizontales et verticales [47]. Cependant, la reconstruction devient possible si on reconstruit des polyominos hv-convexes [5; 18].

Dans ce chapitre on étudie des cas polynomiaux de reconstruction d'images binaires convexes à partir de deux projections.

4.2 Reconstruction polynomiale d'une image binaire convexe

Dans les deux sections suivantes, on fixe une ou deux projections sous forme de vecteurs constants pour résoudre le problème de reconstruction d'images binaires hv-convexes à partir de projections orthogonales.

4.2.1 Reconstruction à partir de deux projections constantes

Dans ce paragraphe, on considère le problème suivant :

$hv - conv(H^{k_1}, V^{k_2})$

Données : $H^{k_1} = (h_1, \dots, h_m) \in \mathbb{N}^m / h_i = k_1; (\forall i)$ et $V^{k_2} = (v_1, \dots, v_n) \in \mathbb{N}^n / v_j = k_2; (\forall j)$.

Sortie : Une image binaire $hv - convexe$, qui respecte H^{k_1} et V^{k_2} .

Proposition 4.2.1 *Si le problème $hv - conv(H^{k_1}, V^{k_2})$ a une solution alors $m = r k_2$ et $n = r k_1$, avec $r \in \mathbb{N}^*$*

Preuve Si le problème $hv - conv(H^{k_1}, V^{k_2})$ a une solution alors :

Le nombre de 1s sur les lignes = $k_1 m$ et le nombre de 1s sur les colonnes = $k_2 n$

D'où $k_1 m = k_2 n \iff m/k_2 = n/k_1 = r$

■

Proposition 4.2.2 *Les 1s de toute solution de $hv - conv(H^{k_1}, V^{k_2})$ forment des blocs de taille $k_2 k_1$.*

Preuve Soit A une solution : Sur la première ligne il y a une barre de longueur k_1 . Supposons que cette barre commence à la cellule $(1, j)$. Ainsi, il y a k_1 barres verticales de longueur k_2 qui commencent entre les cellules $(1, j)$ et $(1, j + k_1 - 1)$. Alors on obtient un premier bloc de taille $k_2 \times k_1$ s'étalant sur les lignes $\{1, \dots, k_2\}$ et les colonnes $\{j, \dots, j + k_1 - 1\}$. Sans perte de généralité, on suppose que le coin supérieur gauche de ce bloc est la cellule $(1, 1)$ (voir Figure 4.1). Après la reconstruction de ce bloc le problème est réduit à reconstruire une image $hv -$

4. Résolution de cas polynomiaux

convexe de taille $(m-k_2, n-k_1)$ avec des projections constantes. Par raisonnement par récurrence, on obtient une solution composée de r blocs.

Le nombre de blocs est $(m k_1)/(k_1 k_2) = (n k_2)/(k_1 k_2) = m/k_2 = n/k_1 = r$

■

	1	2	...	k_1	k_1+1
1	1	1	...	1	
2	1	1	...	1	
⋮	⋮	⋮	⋮	⋮	
k_2	1	1	...	1	
k_2+1					

FIGURE 4.1 – Bloc de taille $k_2 \times k_1$ sur les lignes $\{1, \dots, k_2\}$ et les colonnes $\{1, \dots, k_1\}$

Proposition 4.2.3 *Si $m = r k_2$ et $n = r k_1$, avec $r \in \mathbb{N}^*$, alors le problème $hv - conv(H^{k_1}, V^{k_2})$ a une solution.*

Preuve D'après la proposition 4.2.1 toute solution du problème est composée de $r = m/k_2$ bloc. Une solution possible est la suivante :

- Les coins diagonaux du premier bloc sont $(1, 1)$ et (k_2, k_1)
- Les coins diagonaux du deuxième bloc sont $(1 + k_2, 1 + k_1)$ et $(2 k_2, 2 k_1)$
- Les coins diagonaux du bloc l sont $(1 + (l - 1) k_2, 1 + (l - 1) k_1)$ et $(l k_2, l k_1)$
- Les coins diagonaux du dernier bloc r sont $(1 + (r - 1) k_2, 1 + (r - 1) k_1)$ et (m, n) (voir Figure 4.2).

■

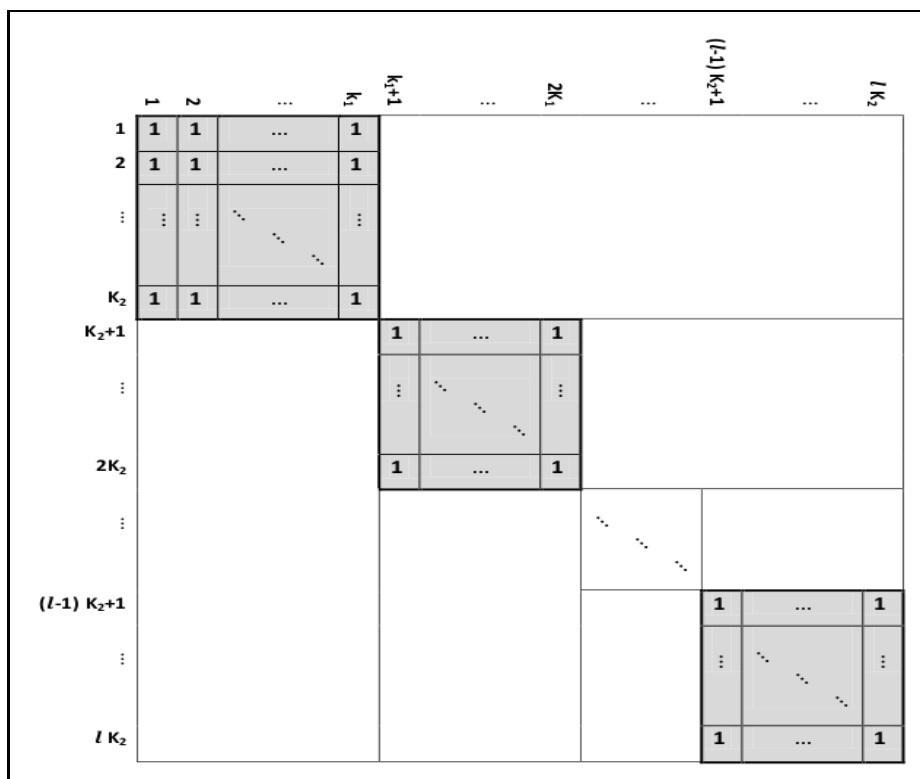


FIGURE 4.2 – Solution du problème $hv - conv(H^{k_1}, V^{k_2})$ est composée de $r = m/k_2$ blocs

Proposition 4.2.4 *Le problème peut avoir $(m/k_2)!$ solutions différentes.*

Preuve Toute solution de $hv - conv(H^{k_1}, V^{k_2})$ est composée de $r = m/k_2$ blocs, c-à-d r positions pour des blocs de taille $k_2 k_1$.

- Le premier bloc peut avoir r emplacements par rapport aux autres blocs de l'image.
- Le deuxième bloc peut avoir $r - 1$ emplacements par rapport aux autres blocs de l'image
- Le l -ème bloc peut avoir $r - l$ emplacements par rapport aux autres blocs de l'image
- Le r -ème bloc (le dernier bloc) peut avoir un seul emplacement par rapport aux autres blocs de l'image,

L'ensemble des solutions de $hv - conv(H^{k_1}, V^{k_2})$ est défini par les différents emplacements de ses r blocs, c-à-d qu'il existe $r \times (r - 1) \times (r - 2) \times \dots \times 2 \times 1$

solutions possibles de $hv - conv(H^{k_1}, V^{k_2})$.

■

Conclusion. La condition $m = r k_2$ et $n = r k_1$, avec $r \in \mathbb{N}^*$ est nécessaire et suffisante pour $hv - conv(H^{k_1}, V^{k_2})$.

4.2.2 Reconstruction à partir d'une seule projection constante

Dans cette section, nous considérons le problème de reconstruction d'une image binaire convexe à partir de projections H et V dont une projection (H ou V) est constante

Le tableau 4.1 résume la complexité des problèmes de reconstruction d'une image binaire hv-convexe avec une et deux projections constantes.

4.2.2.1 Reconstruction d'un polyomino hv-convexe

Dans ce paragraphe, on considère le problème suivant :

$phv - conv(H^k, V)$

Données : $H^k = (h_1, \dots, h_m) \in \mathbb{N}^m / h_i = k; (\forall i)$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Un polyomino $hv - convexe$, qui respecte H^k et V .

Définition 4.2.1 Soit s une ligne, on note par j_s le numéro de colonne du début de la barre horizontale de longueur k sur cette ligne.

Proposition 4.2.5 Les débuts des barres horizontales de toute solution de $phv - conv(H^k, V)$ forment un ensemble monotone.

Preuve Soit S une solution du problème $phv - conv(H^k, V)$. S est un polyomino hv-convexe, alors

- Il existe une seule barre horizontale de longueur k sur chaque ligne i , ($i = 1, \dots, m$).
- Il existe une seule barre verticale de longueur v_j sur chaque colonne j , ($j = 1, \dots, n$).

4. Résolution de cas polynomiaux

On suppose que S est défini par un ensemble non monotone de m barres horizontales. Alors, il existe trois lignes r , s et t telles que : $jr \leq js \geq jt$ ou $jr \geq js \leq jt$ (voir Figure 4.3 et 4.4). On distingue deux cas car toutes les barres horizontales sont de longueur k .

- Si $jr \leq js \geq jt$, alors la colonne $(js - 1)$ est non convexe (il existe des 1s non adjacents verticalement) \implies Contradictoire, voir Figure 4.3.
- Si $jr \geq js \leq jt$, alors la colonne $(js + k + 1)$ est non convexe (il existe des 1s non adjacents verticalement) \implies Contradictoire, voir Figure 4.4.

■

	jr	jt	js	\dots	$jr + k - 1$	$jt + k - 1$	$js + k - 1$	$js + k$
br	1	1	1	\dots	1	0	0	0
bs	0	0	1	\dots	1	1	1	0
bt	0	1	1	\dots	1	1	0	0

FIGURE 4.3 – Trois lignes r , s et t avec : $jr \leq js \geq jt$

	js	jt	jr	\dots	$js + k - 1$	$js + k$	$jr + k - 1$	$jr + k$
br	0	0	1	\dots	1	1	1	0
bs	1	1	1	\dots	1	0	0	0
bt	0	1	1	\dots	1	1	0	0

FIGURE 4.4 – Trois lignes r , s et t avec : $jr \geq js \leq jt$

Remarque 4.2.1 *Le problème $phv - conv(H^k, V)$ peut avoir deux solutions différentes (voir Figure 4.5) selon que les débuts des barres sont croissantes ou*

décroissantes.

<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0
1	1	1	0	0	0	0																																																																															
0	0	1	1	1	0	0																																																																															
0	0	1	1	1	0	0																																																																															
0	0	0	1	1	1	0																																																																															
0	0	0	1	1	1	0																																																																															
0	0	0	0	1	1	1																																																																															
0	0	0	0	1	1	1																																																																															
0	0	0	1	1	1	0																																																																															
0	0	0	1	1	1	0																																																																															
0	0	1	1	1	0	0																																																																															
0	0	1	1	1	0	0																																																																															
1	1	1	0	0	0	0																																																																															

FIGURE 4.5 – Deux solutions différentes du problème $phv - conv(H^3, V)$: $H^3 = (3, 3, 3, 3, 3, 3, 3)$ et $V = (1, 1, 3, 4, 5, 3, 1)$

4.2.2.2 Reconstruction d'un hv-convexe

Le problème de la reconstruction d'une image hv-convexe à partir d'une projection constante est défini par :

$hv - conv(H^k, V)$

Données : $H^k = (h_1, \dots, h_m) \in \mathbb{N}^m / h_i = k; (\forall i)$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$.

Sortie : Une image binaire $hv - convexe$, qui respecte H^k et V .

Définition 4.2.2 Soient $H^k = (h_1, \dots, h_m) \in \mathbb{N}^m / h_i = k$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$, on note par \bar{v}_j : le nombre de barres horizontales commençant à j . Le vecteur $\bar{V} = (\bar{v}_1, \dots, \bar{v}_n) \in \mathbb{N}^n$ est défini par,

$$\bar{V} \begin{cases} \bar{v}_1 = v_1 \\ \bar{v}_2 = v_2 - \bar{v}_1 \\ \bar{v}_j = v_j - \sum_{t=1}^{\max(k-1, 1)} \bar{v}_{j-t}; j = 3, \dots, n \end{cases}$$

Proposition 4.2.6 Deux polyominos hv-convexe adjacents $phv1$ et $phv2$ sont séparés par $(k - 1)$ projections verticales \bar{V} nulles successives.

Preuve Soit $phv1$ et $phv2$ deux polyominos hv-convexes adjacents; $sv1$ et $sv2$ sont respectivement les projections verticales de $phv1$ et $phv2$.

Supposons que les barres horizontales de :

$phv1$ s'étalent entre les colonnes t et t' , $t < t'$,

$phv2$ s'étalent entre les colonnes s et s' , $s < s'$, avec $t' < s$

Pour toute barre horizontale $(b_{i,r})$ de $phv1$ de longueur k , on a :

$(b_{i,r})$ commence à la cellule (i, r) et se termine à la cellule $(i, r+k)$ tel que $r+k \leq s$.

c-à-d, aucune barre horizontale (b_i) de $phv1$ commence à $(i, s - k + 1)$.

Alors : $\sum_{p=s-k+1}^{s-1} \overline{v}_p = 0$

■

Nous déduisons la proposition suivante :

Proposition 4.2.7 *Le nombre de polyominos hv -convexes du problème $hv - conv(H^k, V)$ est égal au nombre de suites de zéros de longueur k dans \overline{V} .*

Preuve Soit $H^k \in \mathbb{N}^m / h_i = k1; (\forall i)$, $V \in \mathbb{N}^n$ deux projections orthogonales du problème $hv - conv(H^k, V)$,

Le vecteur $\overline{V} = (\overline{v}_1, \dots, \overline{v}_n) \in \mathbb{N}^n$ donne le nombre de barres horizontales de $hv - conv(H^k, V)$, voir définition 4.2.2

D'après la proposition 4.2.6 : deux polyominos hv -convexes adjacents sont séparés par $(k - 1)$ projections verticales nulles successives dans \overline{V} .

Alors le nombre de polyominos hv -convexes du problème $hv - conv(H^k, V)$ est donné par le nombre de suites de zéros de longueur $(k - 1)$ dans \overline{V} .

■

4.3 Reconstruction d'une image binaire h-convexe à partir d'une projection verticale

Nous considérons le problème de reconstruction d'une image binaire h-convexe de m lignes à partir d'une projection verticale, noté $h - conv(V, m)$ décrit de la façon suivante :

$h - conv(V, m)$

Données : $V = (v_1, \dots, v_n) \in \mathbb{N}^n$ et $m \in \mathbb{N}$.

Sortie : Une image binaire $h - convexe$ de m lignes, qui respecte V .

Proposition 4.3.1 *Le problème h -conv(V, m) admet une solution si seulement si*

$$m \geq m^* = v_1 + \sum_{j=1}^{n-1} \max\{0, v_{j+1} - v_j\} \quad (1)$$

Preuve Condition nécessaire. En fait le nombre des lignes est supérieur au nombre de barres horizontales car la matrice est h -convexe. Pour chaque colonne j , si $(v_{j+1} > v_j)$ alors au moins $v_{j+1} - v_j$ barres commençant à la colonne $j + 1$.

Condition suffisante. Une solution est entièrement caractérisée par la détermination de :

- s_j le nombre de barres horizontales qui commencent à la colonne j ,
- e_j le nombre de barres horizontales qui se terminent à la colonne j .

L'algorithme suivant donne une solution avec (1) vérifié à l'égalité :

1. $s_1 = v_1$
2. Pour $(j =, \dots, n - 1)$ faire
 - 2.1. Si $(v_{j+1} \leq v_j)$ alors

$$s_{j+1} = 0; e_j = v_j - v_{j+1};$$
 - 2.1. Sinon

$$s_{j+1} = v_{j+1} - v_j; e_j = 0;$$
 - 2.1. Fin Si
3. Fin Pour
4. $e_n = v_n$
5. Fin

Si $m > m^*$, il suffit de mettre à zéro les lignes situées au dessous de la ligne numéro m^* . Le nombre de lignes vaut :

$$m^* = \sum_{j=1}^n s_j = v_1 + \sum_{j=1}^{n-1} \max\{0, v_{j+1} - v_j\}$$

■

Remarque 4.3.1 *Pour que la matrice reconstruite soit aussi v -convexe il suffit de respecter la règle suivante : la barre qui commence la première s'arrête la première. En cas d'égalité, celle qui a le plus petit indice s'arrête la première.*

4.4 Reconstruction d'une image binaire hv-convexe avec projection bornée

Nous considérons le problème de reconstruction, noté $hv - conv(\overline{H}, V)$, suivant :

$hv - conv(\overline{H}, V)$

Données : $\overline{H} = (h_1, \dots, h_m) \in \mathbb{N}^m / h_i \leq k; i = 1 \dots, m$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$, avec $k \in \mathbb{N}^*$.

Sortie : Une matrice binaire hv-convexe, si elle existe, qui respecte \overline{H} et V .

Proposition 4.4.1 *Le problème de reconstruction $hv - conv(\overline{H}, V)$ est NP-complet.*

Preuve Considérons les deux problèmes suivants :

$NMTS(A, B, C)$

Données : $A = (a_1, \dots, a_k) \in \mathbb{N}^k$, $B = (b_1, \dots, b_k) \in \mathbb{N}^k$ et $C = (c_1, \dots, c_k) \in \mathbb{N}^k$

Sortie : Une partition en k triplets de type $(a_i, b_i, c_i) / c_i = a_i + b_i; i = 1, \dots, k$.

$hv - conv(H, V)$

Données : $H = (h_1, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$

Sortie : Une matrice binaire hv-convexe, si elle existe, qui respecte H et V .

Le problème $NMTS(A, B, C)$ est connu pour être NP-Complet. Woeginger [47] a montré que le problème $hv - conv(H, V)$ est aussi NP-Complet en exhibant une réduction polynomiale à partir de problème $NMTS(A, B, C)$. Précisément, la preuve consiste à mettre en équivalence une instance I de $NMTS(A, B, C)$ et une instance I' de $hv - conv(H, V)$ avec $h_i \in \{1, 2\}$. En examinant de près I' , on remarque qu'elle est aussi une instance de $hv - conv(\overline{H}, V)$ (voir figure 4.6).

■

Les tableaux 4.1 et 4.2 ci-dessous résument la complexité des problèmes traités dans ce chapitre :

H													
1	1	0	0	0	0	0	0	0	0	0	0	0	
2	1	1	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	1	1	0	0	0	0	
2	0	0	0	0	0	0	1	1	0	0	0	0	
1	0	0	1	0	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	0	0	
2	0	0	1	1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	1	1	
1	0	0	0	0	1	0	0	0	0	0	0	0	
2	0	0	0	0	1	1	0	0	0	0	0	0	
2	0	0	0	0	1	1	0	0	0	0	0	0	
2	0	0	0	0	1	1	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	1	1	0	0	
	2	1	4	3	5	4	2	2	1	1	1	1	V

FIGURE 4.6 – Réduction polynomiale : Instance du problème $hv - conv(H, V)$ et instance de $NMTS(A, B, C)$, avec $A = \{1, 3, 4\}$; $B = \{2, 1, 1\}$; $C = \{3, 4, 5\}$

4.5 Conclusion

La classe des images hv-convexes est l'une des classes les plus importantes et les plus étudiées en tomographie discrète. Dans le chapitre précédent (chapitre 2) nous avons proposé une méthode fondée sur des techniques de linéarisation et de convexification sur de programmes mathématiques pour des problèmes généraux de tomographie convexe.

Dans ce chapitre nous avons résolu en temps polynomial quelques cas particuliers du problème de reconstruction des images binaires convexes. Nous continuons d'étudier la complexité du problème de reconstruction d'une image binaire avec d'autres contraintes de convexité.

4. Résolution de cas polynomiaux

	(H^k, V^k)	(H^k, V)	(H, V^k)
hv-convexe	P (Section 4.2.1)	P (Section 4.2.2)	P (Section 4.2.2)
Polyomino hv-convexe	P (Section 4.2.1)	P (Section 4.2.2)	P (Section 4.2.2)

TABLE 4.1 – Complexité de problème de reconstruction d’une image binaire convexe : $hv - conv(H^k, V^k)$ et $hv - conv(H^k, V)$ (resp. $hv - conv(H, V^k)$).

	(H, n)	(V, m)	(H, V)	Bornée
h-convexe	P (trivial)	P (Prop. 4.3.1)	NP-C [5]	?
v-convexe	P (Prop. 4.3.1)	P (trivial)	NP-C [5]	?
hv-convexe	P (Prop. 4.3.1)	P (Prop. 4.3.1)	NP-C [5]	NP-C (Prop. 4.4.1)

TABLE 4.2 – Complexité de problème de reconstruction d’une image binaire convexe : $h - conv(V, m)$ (resp. $v - conv(H, n)$) et $hv - conv(H, n)$ (resp. $hv - conv(V, n)$)

4. Résolution de cas polynomiaux

Chapitre 5

Métaheuristiques pour la reconstruction des images binaires

Sommaire

5.1	Introduction	57
5.2	Reconstruction des images binaires hv-convexes	57
5.2.1	Modélisation du problème comme un problème de recherche	58
5.2.2	Méthode de recuit simulé	60
5.2.3	Paramètres de recuit simulé	61
5.3	Résultats numériques	62
5.3.1	Reconstruction $Image_{hv} - conv(H, V)$	62
5.3.2	Interprétations	63
5.4	Conclusion	65

Ce chapitre est basé sur les publications suivantes :

1. A simulated annealing for reconstructing hv-convex binary matrices, *Electronic Notes in Discrete Mathematics* 36 : 447-454 (2010)
2. A simulated annealing approach for reconstructing convex matrices and images from four projections, *In International Conference on Metaheuristics and Nature Inspired Computing, October 2010, Tunisia*

5.1 Introduction

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un grand nombre de problèmes d'optimisation combinatoire. L'objet de ce chapitre est de proposer des méthodes heuristiques pour la reconstruction d'images numériques à partir de leurs projections.

5.2 Reconstruction des images binaires hv-convexes

Le problème de reconstruction d'une image binaire hv-convexe de taille $m \times n$ à partir de projections horizontales $H = (h_1, \dots, h_m)$ et verticales $V = (v_1, \dots, v_n)$ consiste à colorier les pixels d'une image par deux couleurs : noir et blanc (couleur de fond), tel que les projections de l'image donnent le nombre de pixels noirs. Dans cette section, nous considérons le problème de reconstruction d'une image binaire de taille $m \times n$ à partir de projections horizontales et verticales sous une contrainte de convexité.

Définition 5.2.1 (*Convexité*) Soit P une image binaire, on dit que P est

- *h-convexe*, si les pixels noirs de chaque ligne sont adjacents.
- *v-convexe*, si les pixels noirs de chaque colonne sont adjacents.
- *hv-convexe*, si les pixels noirs de chaque ligne et colonne sont adjacents.
- *Polyomino*, si entre chaque paire de pixels noirs il existe un chemin de pixels noirs.

Etant donné une image binaire $P = (p_{i,j})$ de taille $m \times n$ et deux vecteurs $H = (h_1, \dots, h_m)$, $V = (v_1, \dots, v_n)$ à coordonnées entières positives, le problème de reconstruction d'une image binaire hv-convexe à partir de H et V , est défini de la façon suivante :

Image $_hv - conv(H, V)$

Données : $H = (h_1, \dots, h_m) \in \mathbb{N}^m$, $V = (v_1, \dots, v_n) \in \mathbb{N}^n$

Sortie : Une image binaire hv-convexe, si elle existe, qui respecte H et V .

5.2.1 Modélisation du problème comme un problème de recherche

Pour appliquer les approches métaheuristiques à un problème il faut avant tout le transformer en un problème de recherche locale. Cette transformation est réalisée en définissant les critères d'évaluation, la notion de voisinage et la génération de la solution initiale.

5.2.1.1 Fonction objectif

Etant données une image binaire $P = (p_{i,j})$ de taille $m \times n$ et deux vecteurs $H = (h_1, \dots, h_m)$, $V = (v_1, \dots, v_n)$ à coordonnées entières positives. Nous proposons trois critères possibles pour évaluer la qualité des solutions

- a. La première fonction objectif, pour un modèle de maximisation, consiste à calculer le nombre de pixels noirs adjacents dans chaque rangée (ligne et colonne).

$$\max f_1(X) = \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j} x_{i,j+1} + \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j} x_{i+1,j}$$

- * $x_{i,j} \in \{0, 1\}$: indique la couleur du pixel $p_{i,j}$: Si $p_{i,j}$ est noir, alors $x_{i,j} = 1$; sinon $x_{i,j} = 0$.

- b. La deuxième fonction objectif, pour un modèle de minimisation, consiste à calculer le nombre de séquences de pixels noirs dans chaque rangée (une séquence de pixels est une suite successive de pixels noirs horizontaux ou verticaux).

$$\min f_2(X) = \sum_{i=1}^m \sum_{j=1}^n \alpha_{i,j} + \sum_{j=1}^n \sum_{i=1}^m \beta_{i,j}$$

- * $\alpha_{i,j}$ (resp. $\beta_{i,j}$) : indique si une séquence horizontale (resp. verticale) de pixels noirs commence à la ligne i et la colonne j

- c. La troisième fonction objectif consiste à calculer la longueur de la plus

5. Métaheuristiques pour la reconstruction des images binaires

courte séquence de pixels noirs

$$\max f_3(X) = \sum_{i=1}^m r_i + \sum_{j=1}^n c_j$$

* r_i (resp. c_j), la longueur de la plus courte séquence horizontale (resp. verticale) de pixels noirs sur la ligne i (resp. sur la colonne j).

$x_{i,j}, \alpha_{i,j}, \beta_{i,j} \in \{0, 1\}; r_i, c_j \in \mathbb{N}$, pour $i = 1, \dots, m; j = 1, \dots, n$

5.2.1.2 Solution initiale.

Pour calculer une solution initiale, nous utilisons un algorithme glouton basé sur le théorème 2.2.1 de Ryser[42].

Algorithme glouton. A chaque étape (colonne) j , v_j '1' sont placés sur la colonne j et sur les lignes qui ont les plus grandes valeurs de h_i . Si le nombre de lignes ayant les $h_i > 0$ est inférieur à v_j alors l'algorithme s'arrête et il n'y a pas de solution au problème.

Algorithme de Ryser

* Données : $H = (h_1, \dots, h_m) \in \mathbb{N}^m$ et $V = (v_1, \dots, v_n) \in \mathbb{N}^n$.

* Sotie : A , une matrice binaire, qui respecte H et V .

1. Pour ($j = 1, \dots, n$) faire

1.1. Si (il existe v_j lignes ayant $h_i > 0$) alors

Placer (v_j) '1' sur les lignes ayant les (h_i) les plus grands.

Pour tout '1' placé sur une ligne i : $h_i \leftarrow h_i - 1$.

1.2. Sinon, il n'existe pas de solutions. Fin

1.3. Fin si

2. Fin

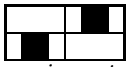
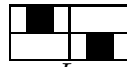
Notons que la solution initiale peut être obtenue par une relaxation linéaire du programme $LP1$ (voir page 29).

5.2.1.3 Voisinage

Ryser [42] a montré que les images binaires, qui respectent les mêmes projections orthogonales H et V , sont équivalentes dans le sens où l'on peut passer

5. Métaheuristiques pour la reconstruction des images binaires

d'une image à une autre par une suite finie d'opérations élémentaires.

Définition 5.2.2 (*Opération élémentaire*) Une bascule est un ensemble de quatre pixels de la forme  ou . Une opération élémentaire consiste à échanger les pixels noirs et blancs. Les projections orthogonales d'une image binaire restent inchangées après les opérations élémentaires.

Théorème 5.2.1 (*Ryser [42]*) Si A et B sont deux images différentes qui respectent les projections orthogonales H et V , alors A se transforme en B (ou inversement), à l'aide d'une suite finie d'opérations élémentaires.

5.2.2 Méthode de recuit simulé

Le recuit simulé (simulated annealing) est souvent présenté comme la plus ancienne des métaheuristiques, la première à mettre spécifiquement en oeuvre une stratégie d'évitement des minima locaux (Kirkpatrick et al. [37]). L'algorithme simule une évolution d'un système physique vers son équilibre thermodynamique à une température T .

Pour reconstruire une image binaire hv-convexe, on part d'une image donnée et on fait subir une modification élémentaire à l'image courante. Si cette perturbation a pour effet d'améliorer une fonction objectif, elle est acceptée. Sinon, elle est acceptée avec une probabilité p . On applique itérativement cette règle un certain nombre de fois à chaque palier L de température pour s'extraire de zones de minima locaux.

En effet, à travers l'algorithme qu'on a proposé, on a essayé de chercher une meilleure solution optimisant (maximiser ou minimiser) une fonction objectif. La reconstruction d'une image hv-convexe à l'aide du recuit simulé se fait en deux étapes :

- Étape 1 : Reconstruire une solution initiale en utilisant une approche gloutonne ou la relaxation linéaire du programme $LP1$ (voir page 29).
- Étape 2 : Choisir une solution voisine de la solution courante.

L'algorithme de recuit simulé est exprimé de la façon suivante pour le critère de maximisation c-à-d la première fonction :

Algorithme de Recuit simulé

Initialisation : $T = 90$, Température initiale ; L , longueur de palier ;

X , une image binaire (solution initiale).

$E = f(X)$, nombre de pixels noirs adjacents pour X .

Répéter L : Si (la Température T est minimale), alors **FIN**

Sinon

Choisir \tilde{X} , une solution équivalente à X

Calculer $\tilde{E} = f(\tilde{X})$, nombre de pixels noirs adjacents pour \tilde{X} .

Si ($\tilde{E} > E$ ou *aléatoire* $< e^{\frac{\tilde{E}-E}{T}}$) alors

$X \leftarrow \tilde{X}$; $E \leftarrow \tilde{E}$;

Diminuer la température T

5.2.3 Paramètres de recuit simulé

L'implémentation de la procédure de recuit simulé nécessite de fixer quatre paramètres, qui jouent un rôle déterminant sur son efficacité.

La température initiale T , dans un premier temps T est choisi très grand, beaucoup de solutions, même celles minimisant la fonction objectif, sont acceptées, et l'algorithme équivaut à une procédure de visite aléatoire de l'espace des solutions. Mais à mesure que la température baisse, la plus part des opérations élémentaires minimisant la fonction objectif sont refusées, et l'algorithme se ramène à une amélioration itérative classique. A température intermédiaire, l'algorithme autorise de temps en temps des opérations qui dégradent la fonction objectif. Il laisse ainsi une chance au système de s'extraire d'un minimum local.

La longueur du palier de température L , qui représente le nombre de solutions testées avant d'appliquer le schéma de décroissance de la température. Si cette longueur est trop faible, l'exploration des solutions est trop faible, alors qu'une valeur trop importante peut ralentir ou même bloquer la recherche.

Le schéma de décroissance de la température, T suit une loi géométrique de décroissance par paliers, $T = T \alpha$, c'est-à-dire que l'on garde T constant, tant

5. Métaheuristiques pour la reconstruction des images binaires

qu'un certain nombre d'opérations élémentaires n'ont pas été réalisées.

Le coefficient α de réduction de la température doit être dans l'intervalle $[0.75, 0.95]$, car une valeur trop faible pour ce coefficient peut provoquer une exploration très partielle de l'espace de recherche.

Comme la méthode de recuit simulé dépend fortement des paramètres, nous avons utilisé les paramètres suivants

- Température initiale, $T = 90$,
- Longueur du palier de température, $L = 10$,
- Le coefficient de réduction de température, $\alpha = 0.9$,
- Loi géométrique de décroissance de température (par palier), $T = T\alpha$,
- Température finale = 5.

5.3 Résultats numériques

Nous présentons dans cette section les résultats numériques obtenus pour le problème de reconstruction d'une image hv-convexe à partir de deux projections. Nous comparons aussi les résultats obtenus par les méthodes suivantes : recuit simulé, programmation linéaire (voir page 29 chapitre 3) et la méthode de Dahl [22].

Nous avons considéré des instances d'images accessibles à partir de : <http://www.inf-uzegedhu/~pbalazs/benchmark/benchmark.html>. Dans la suite ces instances sont groupées en 3 classes (petite, moyenne et grande taille). Chaque instance est notée $(S \times S, C)$, tel que $S \times S$ est la taille de l'image, et C est le nombre de polyominos hv-convexes de l'image à reconstruire.

Tous les algorithmes ont été implémentés en C et exécutés sur un processeur AMD Athlon XP-M 1.7 GHz avec 512 Mo de RAM. Dans toutes les expérimentations, nous avons limité le temps de calcul maximal à 900 seconds.

5.3.1 Reconstruction $Image_{hv-conv}(H, V)$

Pour le problème de reconstruction d'images binaires hv-convexe, nous présentons les résultats de l'algorithme de recuit simulé. Nous comparons aussi les

5. Métaheuristiques pour la reconstruction des images binaires

résultats avec ceux de la méthode de programmation mathématique (voir chapitre 3) et la méthode de Dahl qui a été proposée dans [22].

Les résultats des méthodes de recuit simulé, dahl et programmation linéaire sont présentés dans les tableaux 5.1 et 5.2 et 5.3. Dans les trois tables, la première colonne contient la taille de l'image et le nombre de composantes hv-convexes. Par exemple la ligne (10, 3) concerne l'image de taille 10×10 avec 3 composantes hv-convexes sachant qu'une composante hv-convexe est un polyomino hv-convexe. Les colonnes *Sol*, et *Temps* contiennent respectivement la solution entière et le temps de résolution pour chaque programme : programmation linéaire (noté *LP*), recuit simulé avec une solution initiale calculée par la relaxation linéaire de *LP* (noté *SA – LP*), recuit simulé avec une solution initiale gloutonne (noté *SA – G*) et la méthode de Dahl [22] (noté *DF*).

#image	<i>LP</i>		<i>SA – LP</i>		<i>SA – G</i>		<i>DF</i>	
	Sol	Temps	Sol	Temps	Sol	Temps	Sol	Temps
10 × 10,1	80	0.47	80	0.01	80	0.001	80	0.1
10 × 10,2	36	3.25	36	0.01	36	0.001	36	0.06
10 × 10,3	18	0.62	18	0.12	18	0.002	18	0.06
10 × 10,4	20	0.97	20	0.11	20	0.003	19	0.05
20 × 20,1	390	0.13	390	0.13	390	0.04	390	0.7
20 × 20,2	152	900.03	154	0.14	120	0.02	153	0.6
20 × 20,3	77	900.17	86	0.17	60	0.04	88	0.9
20 × 20,4	45	900.02	45	0.21	39	0.04	45	0.5
30 × 30,1	457	900.02	562	0.22	489	0.05	691	4
30 × 30,2	371	900.06	381	0.23	277	0.04	385	1.8
30 × 30,3	241	900.04	250	0.23	198	0.05	252	2.4
30 × 30,4	273	900.09	300	0.25	271	0.07	315	2.9
40 × 40,1	1177	900.24	1180	0.28	1155	0.16	1189	10
40 × 40,2	627	900.08	663	0.29	420	0.16	732	14.1
40 × 40,3	339	900.22	400	0.34	267	0.18	512	10.8
40 × 40,4	391	900.05	420	0.37	281	0.17	466	18.7

TABLE 5.1 – Résultats de reconstruction pour les images binaires -I-

5.3.2 Interprétations

Nous avons testé les méthodes heuristiques de reconstruction présentées dans ce chapitre et les méthodes présentées dans le chapitre 3 et la méthode de Dahl [22].

5. Métaheuristiques pour la reconstruction des images binaires

#image	<i>LP</i>		<i>SA – LP</i>		<i>SA – G</i>		<i>DF</i>	
	Sol	Temps	Sol	Temps	Sol	Temps	Sol	Temps
50 × 50,1	2578	339.26	2578	1.34	2575	1.58	2577	44.5
50 × 50,2	823	900.34	985	0.37	784	0.42	1162	40.5
50 × 50,3	575	901.20	712	0.27	620	0.25	876	30.3
50 × 50,4	272	900.30	415	0.39	347	0.45	586	21.6
60 × 60,1	2308	900.39	2808	0.43	2732	0.99	2905	246.8
60 × 60,2	1073	900.41	1124	0.84	1115	0.67	1408	92.0
60 × 60,3	515	900.47	627	0.39	573	0.59	1042	86.5
60 × 60,4	491	900.23	600	0.42	519	0.64	839	85.6
70 × 70,1	780	900.07	809	0.65	924	0.86	1052	99.4
70 × 70,2	413	900.12	500	0.81	477	1.09	865	130.9
70 × 70,3	472	900.34	557	0.96	512	0.59	911	139.6
70 × 70,4	575	900.56	679	0.76	480	0.66	1054	155.7

TABLE 5.2 – Résultats de reconstruction pour les images binaires -II-

#image	<i>LP</i>		<i>SA – LP</i>		<i>SA – G</i>		<i>DF</i>	
	Sol	Temps	Sol	Temps	Sol	Temps	Sol	Temps
100 × 100,2	1686	901.13	1804	01.92	1261	1.12	2829	850
120 × 120,2	5466	900.04	5592	04.91	5066	2.12	-	-
140 × 140,2	2347	901.24	2513	11.41	2034	2.29	-	-
160 × 160,2	6471	900.99	6637	05.20	6607	3.93	-	-
180 × 180,2	5680	900.67	5793	07.62	5382	4.93	-	-
200 × 200,2	10195	900.33	10383	19.64	9943	4.66	-	-
220 × 220,2	4207	901.59	4633	12.40	4517	5.66	-	-
240 × 240,2	9997	901.26	11014	09.30	10755	6.24	-	-
260 × 260,2	15754	900.21	16793	19.07	16458	7.89	-	-
280 × 280,2	8048	901.51	8802	44.52	8594	25.10	-	-
300 × 300,2	13319	900.00	13400	45.61	14953	26.08	-	-
320 × 320,2	9758	900.02	9869	29.89	10930	16.10	-	-
340 × 340,2	-	-	-	-	8930	18.91	-	-
360 × 360,2	-	-	-	-	37307	31.67	-	-
380 × 380,2	-	-	-	-	40995	40.37	-	-

TABLE 5.3 – Résultats de reconstruction pour les images binaires -III-; (-) : solution non trouvée au bout de 900s de calcul

5. Métaheuristiques pour la reconstruction des images binaires

Si l'on compare les résultats obtenus par la méthode recuit ($SA - LP$) et recuit ($SA - G$), on voit que la meilleure méthode est la méthode ($SA - LP$). En effet, la valeur de la solution trouvée par $SA - LP$ est bien meilleure que la valeur de la solution calculée par $SA - G$. Globalement, le temps de résolution de $SA - G$ est en moyenne 3 fois plus rapide avec la méthode $SA - LP$. De plus, les deux méthodes permettent de résoudre les instances allant jusqu'à la taille 320×320 en moins de 900 secondes. Autre avantage, la méthode ($SA - G$) permet de résoudre toutes les instances allant jusqu'à la taille 380×380 .

Si l'on compare les résultats obtenus pour les trois méthodes dans les tables 5.1, 5.2 et 5.3, il apparaît que la méthode de Dhal et la méthode de recuit ($SA - G$) sont les meilleures en termes de la valeur de la solution et du temps de reconstruction. Les deux méthodes permettent de reconstruire des images de taille allant de 10×10 à 100×100 en moins de 900 secondes.

5.4 Conclusion

Ce chapitre décrit la résolution de problème de tomographie discrète à l'aide des métaheuristiques. D'une part, nous avons introduit une reformulation du problème de reconstruction comme un problème de recherche. D'autre part, les premiers résultats numériques préliminaires obtenus avec l'algorithme de recuit simulé sont encourageants et des améliorations sur le problème de reconstruction des images pourraient permettre d'obtenir de meilleurs résultats.

Nous envisageons prochainement d'aborder les points suivants :

- amélioration des algorithmes de recuit simulé et recherche tabou,
- développement des métaheuristiques évolutionnaires pour le problème de reconstruction des images,
- approfondissement des propriétés des reformulations obtenues pour la reconstruction des images colorées.

5. Métaheuristiques pour la reconstruction des images binaires

Chapitre 6

Reconstruction des images colorées

Sommaire

6.1	Introduction	68
6.2	Définition du problème	68
6.3	Programmation entière pour la reconstruction d'une image colorée	70
6.4	Méthode de recuit simulé pour la reconstruction des images colorées	73
6.4.1	Fonction objectif	73
6.4.2	Solution initiale	74
6.4.3	Notion de voisinage	75
6.4.4	Algorithme de recuit simulé	77
6.5	Résultats numériques	78
6.5.1	Interprétations	79
6.6	Conclusion	79

Ce chapitre est basé sur les publications suivantes :

1. Combining simulated annealing and linear programming approaches for reconstructing hv-convex colored images, *In The International Symposium on Combinatorial Optimization, April 2012, Athens, Greece*

6.1 Introduction

Nous nous proposons de reconstruire des images colorées à partir de deux projections orthogonales H et V . Pour une image ayant k couleurs, les projections sont définies comme étant le nombre de pixels de chacune des couleurs dans chacune de lignes et colonnes. Le problème est résolu efficacement pour $k = 1$, où chaque pixel est associé à une couleur noire ou blanche. La généralisation de ce problème est NP-complet à partir de $k = 2$ [19; 24; 29].

6.2 Définition du problème

Le problème de reconstruction des images colorées a plusieurs domaines d'applications : la compression de données, la reconstruction d'emploi de temps, l'imagerie médicale, etc. Les projections de départ sont obtenues à l'aide d'un instrument technique, puis un algorithme de reconstruction va fournir une image respectant les projections. Cependant l'image obtenue risque de ne pas ressembler à l'image originale car les projections sont rarement suffisantes pour assurer l'unicité de l'image reconstruite. Ainsi, certaines applications de tomographie exigent d'imposer des informations a priori sur la nature de l'image cherchée (convexité, périodicité, connexité, etc).

Nous rappelons la définition des projections orthogonales (horizontales et verticales) et la convexité pour une image colorée :

Définition 6.2.1 Soient $C = \{1, \dots, k\}$ un ensemble de k couleurs et M une image en couleur de taille $m \times n$ sachant qu'une cellule (i, j) de M est coloriée par une seule couleur $c \in C$, on note par $H = \{H^1, \dots, H^k\}$ les projections horizontales de l'image M et $V = \{V^1, \dots, V^k\}$ les projections verticales de l'image M , tel que :

- $H^c = (h_1^c, \dots, h_m^c) \in \mathbb{N}^m$ est la projection horizontale de la couleur c pour l'image M , avec $h_i^c = |\{j : M_{i,j} = c\}|$ et $c \in C$.
- $V^c = (v_1^c, \dots, v_n^c) \in \mathbb{N}^n$ est la projection verticale de la couleur c pour l'image M , avec $v_j^c = |\{i : M_{i,j} = c\}|$ et $c \in C$.

Plusieurs définitions de la convexité sont possibles.

6. Reconstruction des images colorées

Définition 6.2.2 (*Convexité*) Une image k colorée est

- *h-convexe*, si les pixels de même couleur dans chacune des lignes sont adjacents.
- *v-convexe*, si les pixels de même couleur dans chacune des colonnes sont adjacents.
- *hv-convexe*, si les pixels de même couleur dans chacune des lignes et colonnes sont adjacents.
- *Polyomino*, si entre chaque pairs de pixels de même couleur il existe un chemin de pixels coloré de même couleur.

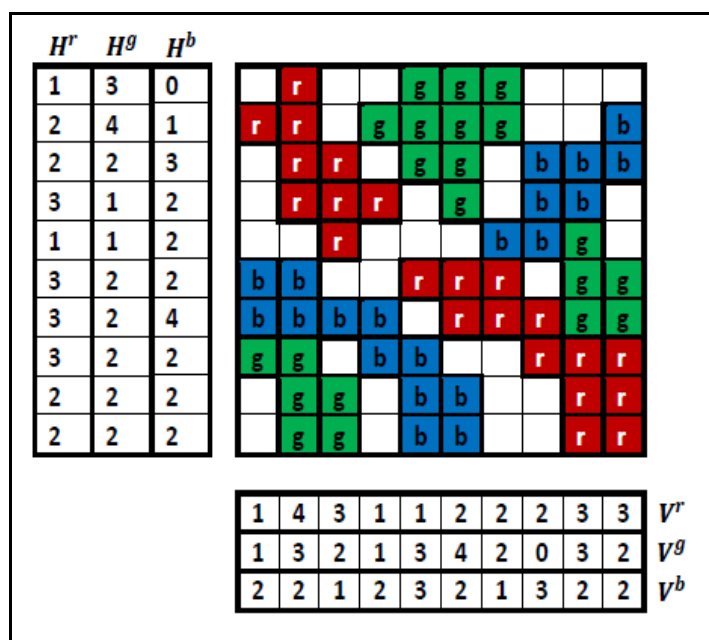


FIGURE 6.1 – Une image 3-colorée convexe avec les projections horizontales et verticales de chaque couleur

Nous considérons le problème de reconstruction d'une image k chromatique convexe suivant :

$$CImage_hv-conv(H^k, V^k)$$

Données : $k \in \mathbb{N}^*$, $H^c = (h_1^c, \dots, h_m^c) \in \mathbb{N}^m$ et $V^c = (v_1^c, \dots, v_n^c) \in \mathbb{N}^n$, pour $c = 1, \dots, k$

Sortie : Une image k chromatique hv-convexe, si elle existe, qui respecte H^c et V^c pour $c = 1, \dots, k$

6.3 Programmation entière pour la reconstruction d'une image colorée

Dans cette section, nous proposons une formulation mathématique en 0 – 1 pour résoudre le problème $CImage_hv-conv(H^k, V^k)$. Nous considérons M une image k chromatique de taille $m \times n$ et $P = \{(H^c, V^c), c = 1, \dots, k\}$ un ensemble de couples de vecteurs entiers positifs représentant les projections horizontales et verticales de l'image M à reconstruire. Nous associons à tout élément (i, j) de M , un vecteur binaire $x_{i,j} = (x_{i,j}^1, \dots, x_{i,j}^k)$ de taille k tel que $x_{i,j}^c = 1$ si l'élément (i, j) est colorié par la couleur c et $x_{i,j}^c = 0$ sinon, avec $c \in \{1, \dots, k\}$. Autrement : $\sum_{c=1}^k x_{i,j}^c \leq 1$ pour tout élément (i, j) de M .

La fonction suivante donne le nombre de 1s adjacents horizontalement et verticalement pour une image colorée de taille $m \times n$:

$$F_{adj}(X) = \sum_{c=1}^k \left(\sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j}^c x_{i,j+1}^c + \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j}^c x_{i+1,j}^c \right)$$

Proposition 6.3.1 *X est une image colorée respectant les projections $H = \{H^1, \dots, H^k\}$ et $V = \{V^1, \dots, V^k\}$, $H^c \in \mathbb{N}^m$ et $V^c \in \mathbb{N}^n$ pour $c = 1 \dots, k$, alors les deux propositions suivantes sont équivalentes :*

1. X est une image colorée hv-convexe
2. $F_{adj}(X) = 2 \sum_{c=1}^k \sum_{i=1}^m h_i^c - k(m+n)$, et $F_{conf}(X) = 0$.

Preuve Soit X , une image colorée qui respecte les projections horizontales et verticales $H = \{H^1, \dots, H^k\}$ et $V = \{V^1, \dots, V^k\}$, $H^c \in \mathbb{N}^m$ et $V^c \in \mathbb{N}^n$ pour $c = 1 \dots, k$.

X est hv-convexe si et seulement si les pixels de même couleur sur chaque ligne $i = 1, \dots, n$ et chaque colonne $j = 1, \dots, m$ sont adjacents, et si tout pixel (i, j) est colorié au plus par une seule couleur, c-a-d :

6. Reconstruction des images colorées

$$\left\{ \begin{array}{l} \sum_{j=1}^{n-1} x_{i,j}^c x_{i,j+1}^c = h_i^c - 1, \text{ pour } i = 1, \dots, m \text{ et } c = 1, \dots, k \\ \sum_{i=1}^{m-1} x_{i,j}^c x_{i+1,j}^c = v_j^c - 1, \text{ pour } j = 1, \dots, n \text{ et } c = 1, \dots, k \\ \sum_{c1=1}^k \sum_{c2=1}^k x_{i,j}^{c1} x_{i,j}^{c2} = 0, \text{ pour } i = 1, \dots, m \text{ et } j = 1, \dots, n. \end{array} \right.$$

Ensuite :

$$\left\{ \begin{array}{l} \sum_{c=1}^k \sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j}^c x_{i,j+1}^c = \sum_{c=1}^k \sum_{i=1}^m h_i^c - k m \\ \sum_{c=1}^k \sum_{j=1}^n \sum_{i=1}^{m-1} x_{i,j}^c x_{i+1,j}^c = \sum_{c=1}^k \sum_{j=1}^n v_j^c - k n \\ \sum_{i=1}^m \sum_{j=1}^n \left(\sum_{c1=1}^k \sum_{c2=1}^k x_{i,j}^{c1} x_{i,j}^{c2} \right) = 0 \end{array} \right.$$

Alors : $\sum_{c=1}^k \left(\sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j}^c x_{i,j+1}^c \right) = 2 \sum_{c=1}^k \sum_{i=1}^m h_i^c - k m - k n$,

Enfin : $F_{adj}(X) = \sum_{c=1}^k \sum_{i=1}^m h_i^c - k(m+n)$, et $F_{conf}(X) = 0$.

■

Le problème $CImage_hv-conv(H^k, V^k)$ est modélisé par le programme, noté IP_{conv} , suivant :

$$IP_{conv} \left\{ \begin{array}{l} \max F_{adj}(X) \\ s.c. \\ \sum_{j=1}^n x_{i,j}^c = h_i^c \quad i = 1, \dots, m; c = 1, \dots, k \quad (1) \\ \sum_{i=1}^m x_{i,j}^c = v_j^c \quad j = 1, \dots, n; c = 1, \dots, k \quad (2) \\ \sum_{c=1}^k x_{i,j}^c \leq 1 \quad i = 1, \dots, m; j = 1, \dots, n \quad (3) \\ x_{i,j}^c \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n; c = 1, \dots, k \end{array} \right.$$

Les contraintes (1) et (2) garantissent que l'image à reconstruire satisfait les projections horizontales et verticales pour chaque couleur c , $c \in \{1, \dots, k\}$. La contrainte (3) assure que tout pixel (i, j) est colorié au plus par une seule couleur. L'objectif de IP_{conv} consiste à maximiser le nombre de pixels adjacents de même couleur.

Nous considérons la linéarisation de IP_{conv} en introduisant les variables binaires $hx_{i,j}^c$ et $vx_{i,j}^c$ pour remplacer les termes quadratiques $x_{i,j}^c x_{i,j+1}^c$ et $x_{i,j}^c x_{i+1,j}^c$ et en ajoutant les inégalités : $2hx_{i,j}^c \leq x_{i,j}^c + x_{i,j+1}^c$ et $2vx_{i,j}^c \leq x_{i,j}^c + x_{i+1,j}^c$ pour assurer l'égalité $hx_{i,j}^c = x_{i,j}^c x_{i,j+1}^c$ et $vx_{i,j}^c = x_{i,j}^c x_{i+1,j}^c$. Le programme linéaire,

6. Reconstruction des images colorées

noté LP_{conv} , équivalent à IP_{conv} est le suivant :

$$LP_{conv} \left\{ \begin{array}{l} \max \sum_{c=1}^k \left(\sum_{i=1}^m \sum_{j=1}^{n-1} hx_{i,j}^c + \sum_{i=1}^{m-1} \sum_{j=1}^n vx_{i,j}^c \right) \\ s.c. \\ \sum_{j=1}^n x_{i,j}^c = h_i^c \quad i = 1, \dots, m; c = 1, \dots, k \quad (1) \\ \sum_{i=1}^m x_{i,j}^c = v_j^c \quad j = 1, \dots, n; c = 1, \dots, k \quad (2) \\ \sum_{c=1}^k x_{i,j}^c \leq 1 \quad i = 1, \dots, m; j = 1, \dots, n \quad (3) \\ 2hx_{i,j}^c \leq x_{i,j}^c + x_{i,j+1}^c \quad (4) \\ 2vx_{i,j}^c \leq x_{i,j}^c + x_{i+1,j}^c \quad (5) \\ hx_{i,j}^c, vx_{i,j}^c, x_{i,j}^c \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n; c = 1, \dots, k \end{array} \right.$$

Pour résoudre le programme LP_{conv} , nous utilisons la relaxation continue de LP_{conv} notée \overline{LP}_{conv} . On commence par résoudre le programme linéaire en continu \overline{LP}_{conv} . Si la solution optimale obtenue pour \overline{LP}_{conv} est entière alors l'optimum pour LP_{conv} est obtenue et le problème $CImage_hv - conv(H^k, V^k)$ est résolu. Autrement, nous utilisons un programme entier Q pour chercher une approximation d'une image colorée z de taille $m \times n$ qui respecte les projections (H^k, V^k) et le plus proche possible de la solution continue du programme \overline{LP}_{conv} . Le programme Q est de la forme suivante :

$$Q \left\{ \begin{array}{l} \max \sum_{c=1}^k \sum_{i=1}^m \sum_{j=1}^n \overline{x}_{i,j}^c z_{i,j}^c \\ s.c. \\ \sum_{j=1}^n z_{i,j}^c = h_i^c \quad i = 1, \dots, m; c = 1, \dots, k \quad (1) \\ \sum_{i=1}^m z_{i,j}^c = v_j^c \quad j = 1, \dots, n; c = 1, \dots, k \quad (2) \\ z_{i,j}^c \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n; c = 1, \dots, k \end{array} \right.$$

Dans l'image z on autorise le chevauchement entre les couleurs. L'image z qui respecte les projections (H^k, V^k) est une approximation de l'image x à reconstruire initialement pour le problème $CImage_hv - conv(H^k, V^k)$. La fonction objectif du programme Q donne le nombre de pixels en commun entre z et \overline{x} . On remarque que le programme Q peut être décomposé en k programmes, $(Q_c)_{c=1, \dots, k}$, avec un programme (Q_c) par couleur. Le programme (Q_c) peut se

reformuler de la façon suivante :

$$Q_c \begin{cases} \max \sum_{i=1}^m \sum_{j=1}^n x_{i,j}^c z_{i,j}^c \\ \text{s.c.} \\ \sum_{j=1}^n z_{i,j}^c = h_i^c \quad i = 1, \dots, m & (1) \\ \sum_{i=1}^m z_{i,j}^c = v_j^c \quad j = 1, \dots, n & (2) \\ z_{i,j}^c \in [0, 1] \quad i = 1, \dots, m; j = 1, \dots, n \end{cases}$$

Le programme Q_c est équivalent au problème de flot max à coût min dans un graphe biparti $G^c = (H^c, V^c, E)$, où $H^c = \{h_i^c, i = 1, \dots, m\}$ représente les lignes et $V^c = \{v_j^c, j = 1, \dots, n\}$ représente les colonnes. Nous ajoutons au graphe G^c un sommet source s et un sommet terminal t (voir Figure 6.2). Les arcs dans le graphe G^c sont regroupés en trois types :

- Un arc de type $(s \rightarrow h_i^c)$ est de capacité h_i^c et de coût 0 ; pour $i = 1, \dots, m$.
- Un arc de type $(h_i^c \rightarrow v_j^c)$ est de capacité 1 et de coût $-x_{i,j}$; pour $i = 1, \dots, m$ et $j = 1, \dots, n$
- Un arc de type $(v_j^c \rightarrow t)$ est de capacité v_j^c et de coût 0 ; pour $j = 1, \dots, n$.

Les arcs de type $(h_i^c \rightarrow v_j^c)$ représentent les pixels de l'image à reconstruire. Q_c a une solution si et seulement si le flot maximal est de valeur $\sum_{i=1}^m h_i^c = \sum_{j=1}^n v_j^c$.

6.4 Méthode de recuit simulé pour la reconstruction des images colorées

Dans cette section, nous proposons une méthode de reconstruction d'une image colorée basée sur l'algorithme de recuit simulé. Pour appliquer cette méthode de reconstruction nous définissons : les critères d'évaluation, la notion de voisinage et la reconstruction d'une solution initiale sont comme suit.

6.4.1 Fonction objectif

Etant donné une image colorée $X = (x_{i,j})$ de taille $m \times n$ et les projections $H = \{H^1, \dots, H^k\}$ et $V = \{V^1, \dots, V^k\}$. Nous évaluons la qualité de la solution

6. Reconstruction des images colorées

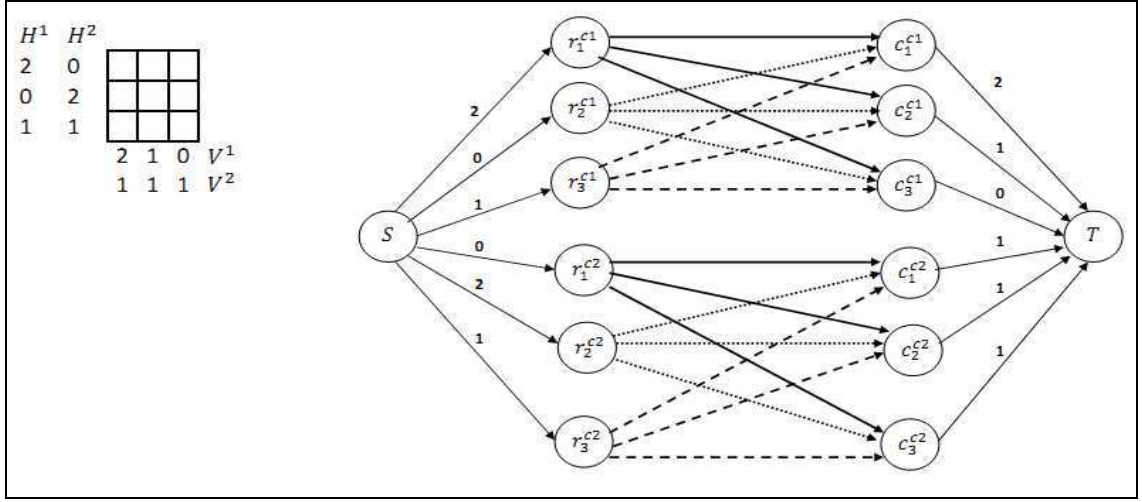


FIGURE 6.2 – Une image colorée et le problème du flot max à coût min dans le graphe associé à $Q_1 : G(H, V, E)$; $H^1 = (2, 0, 1)$; $H^2 = (0, 2, 1)$ et $V^1 = (2, 1, 0)$, $V^2 = (1, 1, 1)$

à l'aide de la fonction objectif F_{adj} et F_{conf} :

- La fonction F_{adj} à maximiser, pour calculer le nombre de pixels adjacents de même couleur horizontalement et verticalement pour une image colorée de taille $m \times n$:

$$F_{adj}(X) = \sum_{c=1}^k \left(\sum_{i=1}^m \sum_{j=1}^{n-1} x_{i,j}^c x_{i,j+1}^c + \sum_{i=1}^{m-1} \sum_{j=1}^n x_{i,j}^c x_{i+1,j}^c \right)$$

- La fonction F_{conf} à minimiser, pour calculer le nombre de conflits entre les pixels de couleur différente pour une image colorée de taille $m \times n$:

$$F_{conf}(X) = \sum_{i=1}^m \sum_{j=1}^n \left(\sum_{c1=1}^k \sum_{c2=1}^k x_{i,j}^{c1} x_{i,j}^{c2} \right)$$

6.4.2 Solution initiale

On admet qu'au cours de la résolution, une cellule peut avoir deux couleurs. Chaque couleur sera codée par une variable.

Pour construire une image colorée A qui respecte les projections horizontales et verticales, nous considérons l'algorithme de Ryser pour reconstruire une image

6. Reconstruction des images colorées

binaire A^c qui respecte les projections H^c et V^c pour la couleur c ($c = 1, \dots, k$). Puis à partir de l'ensemble des images binaires $\{A^1, \dots, A^k\}$, nous utilisons une opération de combinaison (\oplus) définie de la façon suivante : Si A et B sont deux images binaires, alors $S = A \oplus B$ est une image telle que le pixel (i, j) est coloré par la couleur $c1$ (resp. $c2$) si $A_{i,j} = 1$ (resp. $B_{i,j} = 1$). Autrement, si $A_{i,j} = 1$ et $B_{i,j} = 1$ alors (i, j) sera coloré à la fois par $c1$ et $c2$. Alors $A \oplus B = S$ donne :

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} c1 & c2 \\ c1 & c2 \end{pmatrix}$$

Soit l'algorithme de Ryser suivant, pour générer une image binaire A^c qui respecte les projections $H^c = (h_1^c, \dots, h_m^c)$ et $V^c = (v_1, \dots, v_n^c)$, avec $c = 1, \dots, k$:

Algorithme de Ryser

* Données : $H^c = (h_1^c, \dots, h_m^c) \in \mathbb{N}^m$ et $V^c = (v_1, \dots, v_n^c) \in \mathbb{N}^n$.

* Sotie : A^c une image binaire qui respecte H^c et V^c .

1. Pour $i=1$ à n faire

1.1. Si (Il existe v_j^c lignes ayant $h_i^c > 0$) **alors**

Placer v_j^c '1' sur les lignes ayant les (h_i) les plus grands ;

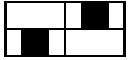

Pour tout '1' placé sur une ligne i : $h_i^c \leftarrow h_i^c - 1$;

1.3. Sinon, il n'existe pas de solutions. Fin ;

2. Fin

6.4.3 Notion de voisinage

Nous rappelons la définition d'une opération élémentaire pour le cas d'une image binaire

Définition 6.4.1 (*Opération élémentaire*) Une bascule est un ensemble de quatre pixels de la forme  ou . Une opération élémentaire consiste à échanger les pixels noirs et blancs. Les projections orthogonales d'une image binaire restent inchangées après les opérations élémentaires.

Ryser [42] a montré que les images binaires, qui respectent les mêmes projections orthogonales H et V , sont équivalentes dans le sens où l'on peut passer d'une image à une autre par une suite finie d'opérations élémentaires.

6. Reconstruction des images colorées

Dans le cas de la reconstruction d'une image colorée, nous proposons la généralisation suivante de la définition d'une opération élémentaire.

Définition 6.4.2 (*Opération élémentaire 'Image Colorée'*) Une bascule est un ensemble de quatre pixels : (i, j) , $(i, j + k)$, $(i + h, j)$ et $(i + h, j + k)$, tel que si on permute les valeurs des éléments de la bascule alors les projections $H = \{H^1, \dots, H^k\}$ et $V = \{V^1, \dots, V^k\}$ restent inchangées (voir table 6.1).

Avant la permutation		Après la permutation									
Nombre de conflits	Bascule	Bascule	Nombre de conflits								
2	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td></td><td>$c_i c_j$</td></tr><tr><td>$c_i c_j$</td><td></td></tr></table>		$c_i c_j$	$c_i c_j$		<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_i</td><td>c_j</td></tr><tr><td>c_j</td><td>c_i</td></tr></table>	c_i	c_j	c_j	c_i	0
	$c_i c_j$										
$c_i c_j$											
c_i	c_j										
c_j	c_i										
1	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td></td><td>c_j</td></tr><tr><td>$c_j c_i$</td><td></td></tr></table>		c_j	$c_j c_i$		<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_j</td><td></td></tr><tr><td>c_i</td><td>c_j</td></tr></table>	c_j		c_i	c_j	0
	c_j										
$c_j c_i$											
c_j											
c_i	c_j										
0	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_i</td><td>c_j</td></tr><tr><td>c_j</td><td>c_i</td></tr></table>	c_i	c_j	c_j	c_i	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_j</td><td>c_i</td></tr><tr><td>c_i</td><td>c_j</td></tr></table>	c_j	c_i	c_i	c_j	0
c_i	c_j										
c_j	c_i										
c_j	c_i										
c_i	c_j										
1	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_t</td><td>c_j</td></tr><tr><td>$c_j c_i$</td><td></td></tr></table>	c_t	c_j	$c_j c_i$		<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>$c_t c_j$</td><td></td></tr><tr><td>c_i</td><td>c_j</td></tr></table>	$c_t c_j$		c_i	c_j	1
c_t	c_j										
$c_j c_i$											
$c_t c_j$											
c_i	c_j										
1	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_t</td><td>c_j</td></tr><tr><td>$c_i c_j$</td><td>c_k</td></tr></table>	c_t	c_j	$c_i c_j$	c_k	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>$c_t c_j$</td><td></td></tr><tr><td>c_i</td><td>$c_j c_k$</td></tr></table>	$c_t c_j$		c_i	$c_j c_k$	2
c_t	c_j										
$c_i c_j$	c_k										
$c_t c_j$											
c_i	$c_j c_k$										
2	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>c_t</td><td>$c_j c_i$</td></tr><tr><td>$c_i c_j$</td><td>c_t</td></tr></table>	c_t	$c_j c_i$	$c_i c_j$	c_t	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>$c_t c_j$</td><td>c_i</td></tr><tr><td>c_i</td><td>$c_j c_t$</td></tr></table>	$c_t c_j$	c_i	c_i	$c_j c_t$	2
c_t	$c_j c_i$										
$c_i c_j$	c_t										
$c_t c_j$	c_i										
c_i	$c_j c_t$										

TABLE 6.1 – Exemples d'opérations élémentaires pour les images colorées

Par la suite deux images colorées A et B sont équivalentes, si :

- A et B respectent les mêmes projections $H = \{H^1, \dots, H^k\}$ et $V = \{V^1, \dots, V^k\}$,
- A se transforme en B (ou inversement), à l'aide d'une suite finie d'opérations élémentaires (voir figure 6.3).

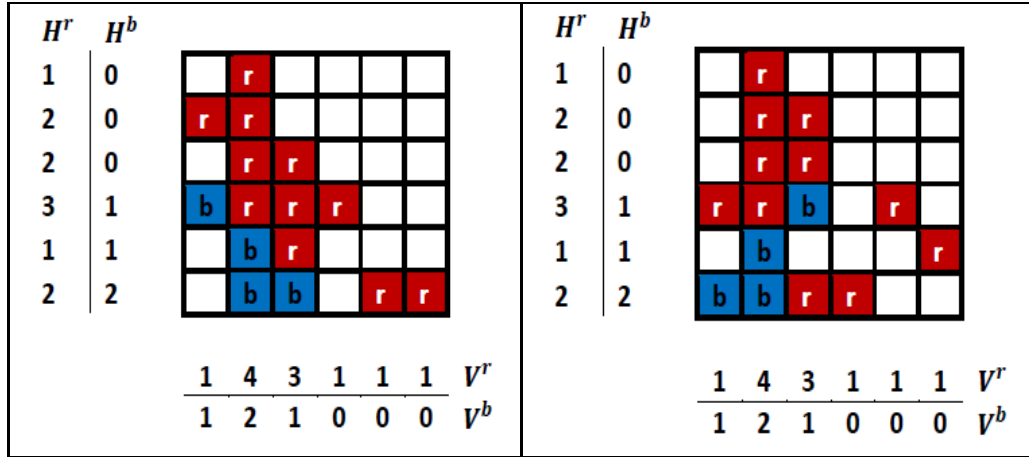


FIGURE 6.3 – Deux images colorées A (à gauche) et B (à droite) équivalentes : à l'aide d'une suite finie d'opérations élémentaires il est possible de transformer A en B ou inversement

6.4.4 Algorithme de recuit simulé

Soit A une image colorée reconstruite par l'algorithme de Ryser, alors A contient des conflits de couleur et n'est pas nécessairement convexe. Pour chercher une image colorée, une solution du problème $CImage_{hv} - conv(H^k, V^k)$, nous considérons simultanément les deux objectifs suivants :

- Premier objectif : Maximiser le nombre de pixels adjacents de même couleur sur chaque ligne et chaque colonne dans l'image A , F_{adj}
- Deuxième objectif : Minimiser le nombre de conflits entre les différentes couleurs dans l'image A , F_{conf}

Pour l'implémentation de l'algorithme de recuit simulé nous considérons les deux objectifs, F_{adj} et F_{conf} , comme un seul objectif. Cette méthode consiste à combiner les deux fonctions F_{adj} et F_{conf} en une seule fonction $F(X) = \alpha F_{adj}(X) - (1 - \alpha)F_{conf}(X)$, où α est une pondération ($0 \leq \alpha \leq 1$). Nous avons utilisé les paramètres suivants :

- Fonction objectif, $F(X) = \alpha_1 F_{adj}(X) - (1 - \alpha_1)F_{conf}(X)$ avec $0 \leq \alpha_1 \leq 1$
- Température initiale, $T = 90$,
- Longueur du palier de température, $L = 10$,
- Le coefficient de réduction de la température, $\alpha = 0.9$,
- Loi géométrique de décroissance de la température, $T = \alpha T$, avec $\alpha = 0.9$,

6. Reconstruction des images colorées

- Nombre maximal d’itérations sans améliorations de l’objectif, $Oper_Max = 100$.
 - Température finale $T_{min} = 5$,
- L’algorithme de recuit simulé est exprimé de la façon suivante :

Algorithme de Recuit simulé

1. *Initialisation* : $T = 90$, Température initiale ; L , longueur de palier ; $\alpha \in [0, 1]$;
 X , une image colorée (solution initiale).
 $E = F(X) = \alpha F_{adj}(X) - (1 - \alpha) F_{conf}(X)$.
2. *Répéter L fois* : Si (la Température T est minimale), alors **FIN**
Sinon
 - a. Choisir \tilde{X} , une solution équivalente à X
 - b. Calculer $\tilde{E} = F(\tilde{X})$.
 - c. Si ($\tilde{E} > E$ ou $rand < e^{\frac{\tilde{E}-E}{T}}$) alors
 - i. $X \leftarrow \tilde{X}$;
 - ii. $E \leftarrow \tilde{E}$;
3. *Diminuer la température T , et retourner en 2*

6.5 Résultats numériques

Pour évaluer les méthodes de reconstruction : programmation linéaire et recuit simulé, nous utilisons d’images colorées convexes générées à partir d’un ensemble d’images binaires convexes générées par la méthode décrite dans [2; 3; 4]. Dans l’étude numérique, le temps d’exécution et la qualité de la solution finale, exprimée par le nombre de paires adjacentes de couleurs F_{adj} et le nombre de conflits entre les couleurs F_{conf} , sont les deux mesures de comparaison entre les deux méthodes.

Les résultats obtenus sont résumés dans les tableaux 6.2, 6.3, 6.4 et 6.5. Dans ces tableaux, la colonne "taille" donne la taille, le nombre de couleurs et le nombre de composants hv-convexes pour chaque couleur ; un composant est un polyomino hv-convexe maximal, et chaque image peut être décomposée en un nombre minimum de composants par couleur. Par exemple, la ligne (30, 3, 2) exprime que la taille de l’image est 30×30 , qu’elle comprend trois couleurs et deux composants pour chaque couleur. La deuxième colonne donne le nombre maximal d’élé-

ments adjacents (F^*). La colonne étiquetée F_{adj} (respectivement F_{Conf}) contient le nombre d'éléments adjacents (resp. le nombre de conflits entre les couleurs). La colonne de temps contient le temps CPU d'exécution (en secondes) requis par chaque méthode. Comme le recuit est un algorithme stochastique, nous avons répété chaque test 10 fois et calculé la moyenne des 10 valeurs obtenus.

6.5.1 Interprétations

Nous comparons la reconstruction heuristique pour les images dont la taille est comprise entre 10×10 et 40×40 . Nous notons que généralement la méthode de programmation linéaire donne une reconstruction bien meilleure que la méthode de recuit. Même si le temps nécessaire pour reconstruire des images de petite taille est négligeable pour les deux méthodes, on constate que la méthode de recuit est la plus rapide.

Nous comparons les méthodes de reconstruction pour d'images dont la taille est comprise entre 50×50 et 80×80 . Les résultats montrent une augmentation du temps de reconstruction, et on remarque que la méthode de recuit donne plus de conflits que la méthode de programmation linéaire (LP). Notons que la contrainte d'exclusivité a été relâchée en utilisant l'algorithme de Ryser.

La figure 6.4 illustre un exemple de reconstruction approchée par le recuit simulé à l'aide d'une solution initiale calculée par la méthode de programmation linéaire LP .

6.6 Conclusion

Dans ce chapitre, nous avons considéré le problème de la reconstruction des images colorées comme une généralisation de problème standard de tomographie discrète. Nous avons formulé ce problème à l'aide d'un programme quadratique en nombres entiers et nous avons proposé une linéarisation pour ce modèle. Pour la résolution, nous avons utilisé deux méthodes d'approximation : une première méthode basée sur le problème de flot max à coût min dans un graphe biparti, et une deuxième méthode basée sur l'algorithme de recuit simulé.

6. Reconstruction des images colorées

size	F^*	SA-LP			SA-G		
		F_{adj}	Conf	Time	F_{adj}	Conf	Time
(10,2,1)	54	54	0	0.22	54	0	0.40
(10,2,2)	78	78	0	0.31	78	0	0.29
(10,2,3)	46	46	0	0.39	46	0	0.38
(10,2,4)	40	40	0	0.42	40	0	0.42
(10,3,1)	65	65	0	0.21	65	0	0.17
(10,3,2)	63	63	0	0.44	63	0	0.52
(10,3,3)	62	62	0	0.15	62	0	0.40
(10,3,4)	46	46	0	0.32	46	0	0.31
(10,4,1)	56	56	0	0.48	56	0	0.52
(10,4,2)	32	32	0	0.39	32	0	0.40
(10,4,3)	67	67	0	0.50	67	0	0.63
(10,4,4)	49	49	0	0.45	49	0	0.83
(20,2,1)	402	402	0	6.25	402	0	8.3
(20,2,2)	294	294	0	7.23	294	0	7.5
(20,2,3)	176	176	0	8.31	176	0	7.9
(20,2,4)	172	172	0	6.37	40	0	9.9
(20,3,1)	231	231	0	9.20	231	0	8.2
((20,3,2)	175	175	0	8.54	175	0	9.3
(20,3,3)	62	62	0	8.01	62	0	10.02
(20,3,4)	227	227	0	9.41	227	0	9.01
(20,4,1)	360	360	0	7.69	360	0	8.72
(20,4,2)	213	213	0	8.65	213	0	10.73
(20,4,3)	294	294	0	6.72	294	0	9.55
(20,4,4)	235	235	0	7.45	235	0	10.05

TABLE 6.2 – Résultats numériques pour la reconstruction des images colorées, ensemble 1

6. Reconstruction des images colorées

size	F^*	SA-LP			SA-G		
		F_{adj}	Conf	Time	F_{adj}	Conf	Time
(30,2,1)	419	419	0	15.60	419	0	18.20
(30,2,2)	634	634	0	16.81	634	0	19.33
(30,2,3)	518	518	0	15.75	518	0	17.41
(30,2,4)	346	346	0	17.49	346	0	20.63
(30,3,1)	536	536	0	14.78	536	0	16.83
(30,3,2)	350	350	0	13.22	350	0	15.15
(30,3,3)	482	482	0	16.33	482	0	16.00
(30,3,4)	543	543	0	15.52	543	0	15.00
(30,4,1)	663	663	0	15.69	663	0	22.21
(30,4,2)	352	352	0	16.99	352	0	19.45
(30,4,3)	643	643	0	18.34	643	0	20.33
(30,4,4)	542	542	0	16,56	542	0	18.22
(40,2,1)	928	928	0	20.03	928	0	24.22
(40,2,2)	1056	1056	0	22.42	1056	0	25.92
(40,2,3)	700	700	0	23.12	700	0	23.66
(40,2,4)	602	602	0	21.61	602	0	21.89
(40,3,1)	803	803	0	22.00	803	0	25.23
(40,3,2)	833	883	0	24.38	883	0	27.04
(40,3,3)	836	836	0	23.65	836	0	28.13
(40,3,4)	532	532	0	22.10	532	0	30.01
(40,4,1)	1096	1096	0	25.73	1096	0	30.11
(40,4,2)	1007	1007	0	24.54	1007	0	29.20
(40,4,3)	1180	1180	0	26.83	1180	0	30.78
(40,4,4)	994	994	0	27.01	994	0	29.71

TABLE 6.3 – Résultats numériques pour la reconstruction des images colorées, ensemble 2

6. Reconstruction des images colorées

size	F^*	SA-LP			SA-G		
		F_{adj}	Conf	Time	F_{adj}	Conf	Time
(50,2,1)	1024	1020	0	40.22	2006	403	51.80
(50,2,2)	1228	1207	0	42.08	2195	533	54.07
(50,2,3)	1258	1220	0	43.58	2434	647	57.42
(50,2,4)	990	951	0	45.90	3761	565	58.42
(50,3,1)	1076	996	0	49.67	933	0	58.74
(50,3,2)	1379	1261	0	48.32	943	2	53.63
(50,3,3)	1540	1425	0	42.49	1024	3	56.29
(50,3,4)	1612	1487	0	50.01	1007	9	59.10
(50,4,1)	1813	1758	0	52.84	1467	0	63.11
(50,4,2)	966	908	0	50.00	704	0	60.48
(50,4,3)	876	701	0	55.10	531	0	68.51
(50,4,4)	1054	662	0	51.09	494	0	62.09
(60,2,1)	1217	1123	0	69.12	3208	928	80.12
(60,2,2)	2484	1618	0	69.08	3162	1371	83.51
(60,2,3)	1972	1289	0	70.29	4040	1400	85.00
(60,2,4)	1632	1089	0	74.53	4732	1783	84.91
(60,3,1)	1574	1441	0	72.89	1051	0	89.71
(60,3,2)	2839	2056	0	79.02	1495	11	90.80
(60,3,3)	2603	1687	0	80.12	1097	18	93.09
(60,3,4)	2070	1287	0	85.83	928	2	95.02
(60,4,1)	1244	1150	0	81.80	807	0	98.39
(60,4,2)	2002	1393	0	80.64	1007	0	101.33
(60,4,3)	2411	1364	0	83.42	1057	0	102.29
(60,4,4)	1216	897	0	85.01	622	0	103.73

TABLE 6.4 – Résultats numériques pour la reconstruction des images colorées, ensemble 3

6. Reconstruction des images colorées

size	F^*	SA-LP			SA-G		
		F_{adj}	Conf	Time	F_{adj}	Conf	Time
(70,2,1)	1350	1173	0	90.01	5183	949	120
(70,2,2)	2145	1301	0	92.14	6097	1120	122
(70,2,3)	2055	1204	0	99.72	5808	1320	123
(70,2,4)	2128	1100	0	103.09	7786	1601	125
(70,3,1)	3026	2967	0	100.21	2682	0	128.11
(70,3,2)	3037	2401	0	109.20	1434	12	123.21
(70,3,3)	3434	1975	0	110.67	1328	12	126.43
(70,3,4)	2706	1650	0	112.00	1350	8	129.51
(70,4,1)	2794	2692	0	108.22	2364	0	132.36
(70,4,2)	2771	1989	0	103.09	1612	0	131.72
(70,4,3)	2409	1449	0	115.38	1012	0	136.81
(70,4,4)	2220	1217	0	121.82	1081	0	139.06
(80,2,1)	2172	1605	0	110.02	5648	1649	120.63
(80,2,2)	9374	2580	0	112.31	6085	3080	125.21
(80,2,3)	3178	2269	0	119.09	7125	3326	128.81
(80,2,4)	2352	1657	0	121.82	7888	3539	132.61
(80,3,1)	3042	2059	0	121.32	2001	0	130.41
(80,3,2)	4929	3010	0	125.54	3265	13	139.52
(80,3,3)	4160	2629	0	131.44	1969	12	140.71
(80,3,4)	3719	2308	0	129.19	1661	10	149.13
(80,4,1)	2747	2633	0	134.33	2443	0	151.04
(80,4,2)	3284	1677	0	139.80	2429	0	154.19
(80,4,3)	3663	1928	0	140.62	1260	0	152.52
(80,4,4)	3590	1798	0	141.11	1573	1	155.43

TABLE 6.5 – Résultats numériques pour la reconstruction des images colorées, ensemble 4

6. Reconstruction des images colorées

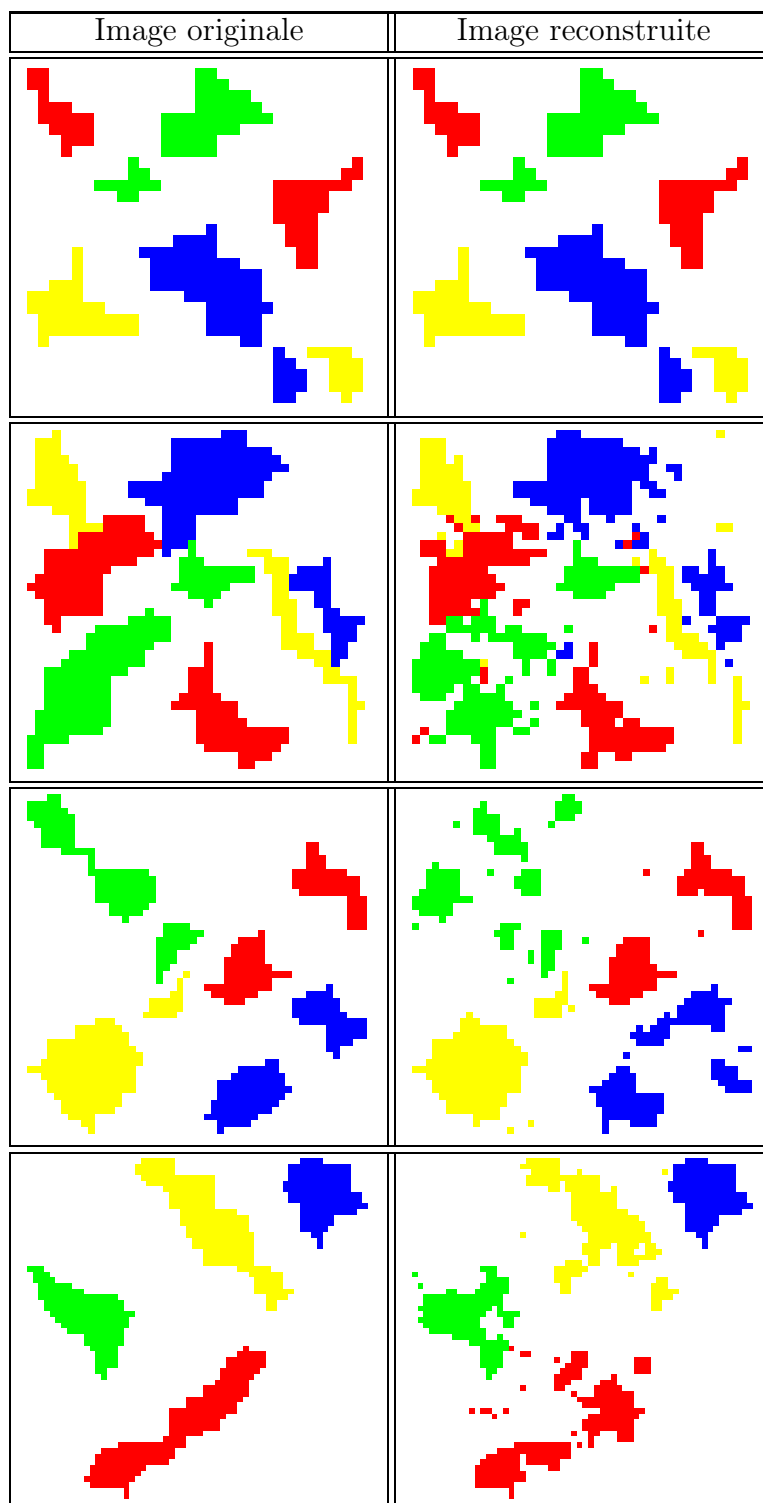


FIGURE 6.4 – Exemple de reconstruction des images colorées

6. Reconstruction des images colorées

Chapitre 7

Conclusion Générale

Ce travail de thèse porte sur l'utilisation de techniques d'optimisation combinatoire pour le problème de tomographie discrète. L'objectif principal consiste à proposer, des méthodes de reconstruction tomographique des images binaires et colorées fondées sur les techniques de la programmation mathématique.

Tout d'abord, le premier chapitre est consacré à la tomographie discrète d'une manière générale et à la reconstruction des matrices binaires en particulier. Nous avons présenté un état de l'art des différents aspects abordables en tomographie : algorithmique, combinatoire, géométrique, etc. Puis, nous avons présenté l'utilisation de la tomographie discrète dans plusieurs domaines : l'imagerie médicale, planification de personnel, microscopie électronique, etc. Nous avons ensuite présenté quelques méthodes de reconstruction des matrices binaires. En particulier pour les méthodes de programmation mathématique, nous avons expliqué la méthode de Dahl et Flatberg. Nous avons discuté aussi les classes les plus importantes et les plus étudiées en tomographie discrète en citant quelques classes de reconstruction tomographique polynomiales.

Le deuxième chapitre introduit la modélisation du problème de tomographie discrète à l'aide des formulations mathématiques en nombres entiers. Nous avons proposé des méthodes de linéarisation et de convexification pour rendre les programmes proposés efficaces. Ainsi, une étude concernant la formulation et les propriétés de l'objet à reconstruire a été réalisée. Nous avons également effectué une expérimentation sur la reconstruction des images binaires par des programmes linéaires et des programmes convexes.

Principalement en tomographie discrète il existe deux classes polynomiales :

- La reconstruction d’une matrice binaire à partir de deux projections.
- La reconstruction d’une matrice binaire connexe et convexe.

Le chapitre 4 est consacré à la résolution de cas particuliers de reconstruction d’images convexes. Nous avons montré que la reconstruction d’une image convexe de projections constantes est un problème polynomial. Nous avons déduit, du théorème de Wognegier, que la reconstruction d’une image hv-convexe à partir de deux projections orthogonales bornées est NP-complet.

Le quatrième chapitre de notre travail étudie expérimentalement des métaheuristiques pour la reconstruction tomographique des images binaires. Nous avons constaté que la méthode de recuit simulé combinée avec la programmation linéaire permet d’accélérer significativement la reconstruction tomographique et d’étendre la reconstruction à des images de grande taille.

Finalement, le chapitre 6 généralise le problème de reconstruction de matrices binaires. Il s’agit de colorier les cellules d’une matrice, à raison d’une seule couleur par cellule en respectant les contraintes données sur le nombre de couleurs dans chaque ligne et chaque colonne. En effet, cette généralisation est envisageable dans le cadre de plusieurs applications de tomographie discrète comme l’imagerie médicale, la localisation des objets en mouvement (RTI : Radio Tomographic Imaging), la microscopie électronique, etc. Nous avons commencé par modéliser mathématiquement cette généralisation. Puis, nous avons abordé la reconstruction des images colorées à l’aide de métaheuristiques d’une part et de la programmation linéaire en nombres entiers d’autre part. Enfin, nous avons présenté les résultats expérimentaux associés.

La thématique principale de cette thèse est la tomographie. Elle est définie principalement par deux processus, un processus d’acquisition pour calculer les projections et un processus de reconstruction. En algorithmique on s’intéresse plutôt à l’aspect reconstruction. Ce processus de reconstruction est extrêmement difficile pour plusieurs raisons : les projections sont en nombre limité, la non homogénéité de l’objet à reconstruire, etc. Nous pouvons conclure, que les méthodes d’optimisation combinatoire sont efficaces pour la reconstruction des objets convexes. Cependant, par manque de temps, nous n’avons pas pu utiliser ces méthodes pour la reconstruction des objets non homogènes avec un grand nombre

7. Conclusion Générale

de projections. Cela fait partie des recherches futures.

7. Conclusion Générale

Références

- [1] P. Balázs. *Binary Tomography Using Geometrical Priors : Uniqueness and Reconstruction Results*. PhD thesis, University of Szeged, 2007.
- [2] P. Balázs. A framework for generating some discrete sets with disjoint components by using uniform distributions. *Discrete Applied Mathematics*, 406 :15–23, 2008.
- [3] P. Balázs. A benchmark set for the reconstruction of hv-convex discrete sets. *Discrete Applied Mathematics*, 157 :3447–3456, 2009.
- [4] P. Balázs. Benchmark collections for the reconstruction of hv-convex discrete sets : <http://www.inf-u-szeged.hu/~pbalazs/benchmark/benchmark.html>, 2010.
- [5] E. Barcucci, A. Del Lungo, M. Nivat, and R. Pinzani. Reconstructing convex polyominoes from their horizontal and vertical projections. *Theoretical Computer Science*, 155 :321–347, 1996.
- [6] K. J. Batenburg. *Network Flow Algorithms for Discrete Tomography*. PhD thesis, Leiden University. The Netherlands, 2006.
- [7] K.J. Batenburg and W.J. Palenstijn. On the reconstruction of crystals through discrete tomography. *Lecture Notes in computer science*, 3322 :23–37, 2004.
- [8] R.J. Baxter. Planar lattice gases with nearest-neighbour exclusion. *Annals of Combinatorics*, 3 :191–203, 1999.

- [9] A. Billionnet. Optimisation quadratique en variables 0-1. In *Optimisation combinatoire, concepts fondamentaux*, pages 191–234. Lavoisier, 2005.
- [10] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109 :55–68, 2007.
- [11] A. Billionnet, S. Elloumi, and A. Lambert. Extending the qcr method to general mixed-integer programs. *Mathematical Programming*, 131 :381–401, 2012.
- [12] A. Billionnet, S. Elloumi, and Marie-Christine Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation : The qcr method. *Discrete Applied Mathematics*, 157 :1185–1197, 2009.
- [13] E. Boros and P.L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123 :155–225, 2002.
- [14] Y. Boufkhad, O. Dubois, and M. Nivat. Reconstructing (h; v)-convex 2-dimensional patterns of objects from approximate horizontal and vertical projections. *Theoretical Computer Sciences*, 290 :1647–1664, 2003.
- [15] S. Brocchi, A. Frosini, and C. Picouleau. Reconstruction of binary matrices under fixed size neighborhood constraints. *Theoretical Computer Science*, 406 :43–54, 2008.
- [16] S. Brunetti. *Convexity and Complexity in Discrete Tomography*. PhD thesis, University of Florence. Italy, 2001.
- [17] S. Brunetti, M. C. Costa, A. Frosini, F. Jarray, and C. Picouleau. Reconstruction of binary matrices under adjacency constraints. In *Advances in Discrete Tomography and its Applications*, page 125–150. Birkhauser, 2007.
- [18] M. Chrobak and Ch. Dürr. Reconstructing hv-convex polyominoes from orthogonal projections. *Information Processing Letters*, 69 :283–289, 1999.

- [19] M. Chrobak and Ch. Dürr. Reconstructing polyatomic structures from x-rays : Np-completeness proof for three atoms. *Theoretical computer science*, 259 :81–98, 2001.
- [20] M. C. Costa, D. de Werra, and C. Picouleau. Using graphs for some discrete tomography problems. *Discrete Applied Mathematics*, 154 :35–46, 2006.
- [21] M. C. Costa, F. Jarray, and C. Picouleau. Reconstruction of binary matrices under adjacency constraints. *Electronic Notes in Discrete mathematics*, 20 :281–197, 2005.
- [22] G. Dahl and T. Flatberg. Optimization and reconstruction of hv-convex (0,1)-matrices. *Discrete Applied Mathematics*, 151 :93–105, 2005.
- [23] A. Daurat. *Convexité dans le Plan Discret. Application à la Tomographie*. PhD thesis, Université Paris 7. France, 2000.
- [24] Ch. Dürr, F. Guinez, and M. Matamala. Reconstructing 3-colored grids from horizontal and vertical projections is np-hard. In *Lecture Notes in Computer Science*, volume 5757, pages 776–787. Proceedings of the 17th Annual European Symposium on Algorithms (ESA), 2009.
- [25] P. Flajolet, P Zimmerman, and B. Van Cusern. A calculus for random generation of labeled combinatorial structures. *Theoretical Computer Sciences*, 132 :1–35, 1994.
- [26] A. Frosini, C. Picouleau, and S. Rinaldi. Reconstructing binary matrices with neighborhood constraints : An np-hard problem. In *Lecture Notes in Computer Science*, pages 392–400, 2008.
- [27] D. Gale. A theorem on flows in networks. *Pacific Journal Mathematics*, 7 :1073–1082, 1957.
- [28] R.J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational complexity of reconstructing lattice sets from their x-rays. *Discrete Mathematics*, 202 :45–71, 1999.

- [29] R.J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational complexity of determining polyatomic structures by x-rays. *Theoretical computer science*, 233 :91–106, 2000.
- [30] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math. Soc.*, 64 :275–278, 1958.
- [31] R. H. Haber. Term rank of 0-1 matrices. Technical Report 30 : 24-51, Univ Padova, 1960.
- [32] G. T. Herman and A. Kuba, editors. *Discrete Tomography : Foundations, Algorithms and Applications*. Birkhauser, 1999.
- [33] G. T. Herman and A. Kuba. Discrete tomography in medical imaging. *Proceedings Of The IEEE*, 94 :1612–1626, 2003.
- [34] F. Jarray. *Résolution de problèmes de tomographie discrète. Applications à la planification de personnel*. PhD thesis, CNAM. Paris. France, 2007.
- [35] F. Jarray, M. C. Costa, and C. Picouleau. Approximating hv-convex binary matrices and images from discrete projections. In *Lecture Notes in Computer Science*, pages 413–422, 2008.
- [36] F. Jarray and G. Tlig. A simulated annealing for reconstruction hv-convex binary matrices. In *Electronic Notes in Discrete mathematics*, pages 413–422, 2010.
- [37] S. Kirkpatrick, C. D. Jr Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 :671–680, 1983.
- [38] A. Del Lungo, A. Frosini, M. Nivat, and L. Vuillon. Discrete tomography : reconstruction under periodicity constraints. *Lecture Notes in Computer Science*, 2380 :38–56, 2002.
- [39] M. L. Mehta. On the problem of hard squares. *The Journal of Chemical Physics*, 60 :2207–2209, 1974.
- [40] M. Nivat. Tomographie discrète. In *Colloquium Jacques Morgenstern*, 2003.

-
- [41] C. Picouleau. Reconstruction of domino tiling from its two orthogonal projections. *Theoretical Computer Science*, 255 :437–447, 2001.
- [42] H. J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canada Journal Mathematics*, 9 :371–377, 1957.
- [43] K. Tawbe and L. Vuillon. 2l-convex polyominoes : Geometrical aspects. *Contributions to Discrete Mathematics*, 6, 2011.
- [44] W. van Aarle. *Tomographic segmentation and discrete tomography for quantitative analysis of transmission tomography data*. PhD thesis, University of Antwerp, Belgium, 2012.
- [45] B. Wang and F. Zhang. On the precise number of $(0,1)$ -matrices in $u(r,s)$. *Discrete Mathematics*, 187 :211–220, 1998.
- [46] J. Wilson and N. Patwari. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing*, Vol 9, No 5 :621–632, May 2010.
- [47] G. J. Woeginger. The reconstruction of polyominoes from their orthogonal projections. *Information Processing Letters*, 77 :225–229, 2001.
- [48] Y. Zhao, N. Patwari, J. M. Phillips, and S. Venkatasubramanian. Radio tomographic imaging and tracking of stationary and moving people via kernel distance. *Information Processing in Sensor Networks*, 2013.

Bibliographie de l'auteur

Cette bibliographie rassemble les publications de l'auteur, dans le domaine de la tomographie discrète, cités par ordre chronologique,

1. A. Billionnet, F. Jarray, G. Tlig and E. Zagrouba : **Reconstruction des matrices binaires sous contrainte de voisinage**. *Congrès ROADEF, Février 2013 Troyes, France.*
2. A. Billionnet, F. Jarray, G. Tlig and E. Zagrouba : **Reconstructing convex matrices by integer programming approaches**. *Journal of Mathematical Modelling and Algorithms 2012.*
3. A. Billionnet, F. Jarray, G. Tlig and E. Zagrouba : **Combining simulated annealing and linear programming approaches for reconstruction hv-convex colored images**. *In the International Symposium on Combinatorial Optimization, April 2012 Athens, Greek.*
4. A. Billionnet, F. Jarray, G. Tlig and E. Zagrouba : **Programmation discrète pour la reconstruction de matrices convexes**. *Congrès ROADEF, Avril 2012 Angers, France.*
5. A. Billionnet, F. Jarray, G. Tlig and E. Zagrouba : **Local search algorithm for reconstructing homogeneous binary images**. *The 4rd International Conference on Metaheuristics and Nature Inspired Computing - META'12 Tunisia, October 2012.*
6. F. Jarray and G. Tlig : **Approximating bicolored images from discrete projections**. *In 14th International workshop on Combinatorial Images Analysis, May 2011, Madrid, Espagne. Lectures Notes in Computer Sciences (LNCS)*

-
7. F. Jarray and G. Tlig : **A simulated annealing for reconstruction hv-convex binary matrices.** *10th International Symposium on combinatorial optimization - ISCO 2010 Tunisia, Mars 2010.* Electronic Notes in Discrete Mathematics (ENDM), 36 (1), 447-454, 2010
 8. F. Jarray and G. Tlig : **A simulated annealing approach for reconstructing convex matrices and images from four projections.** *The 3rd International Conference on Metaheuristics and Nature Inspired Computing - META'10 Tunisia, October 2010.*
 9. F. Jarray and G. Tlig : **Reconstructing hv-convex images by tabu research approach.** *The 3rd International Conference on Metaheuristics and Nature Inspired Computing - META'10 Tunisia, October 2010.*



Ghassen TLIG

Programmation Mathématique en Tomographie Discrète

Résumé

La tomographie est un ensemble de techniques visant à reconstruire l'intérieur d'un objet sans toucher l'objet lui-même comme dans le cas d'un scanner. Les principes théoriques de la tomographie ont été énoncés par Radon en 1917. On peut assimiler l'objet à reconstruire à une image, matrice, etc.

Le problème de reconstruction tomographique consiste à estimer l'objet à partir d'un ensemble de projections obtenues par mesures expérimentales autour de l'objet à reconstruire. La tomographie discrète étudie le cas où le nombre de projections est limité et l'objet est défini de façon discrète. Les champs d'applications de la tomographie discrète sont nombreux et variés. Citons par exemple les applications de type non destructif comme l'imagerie médicale. Il existe d'autres applications de la tomographie discrète, comme les problèmes d'emplois du temps.

La tomographie discrète peut être considérée comme un problème d'optimisation combinatoire car le domaine de reconstruction est discret et le nombre de projections est fini. La programmation mathématique en nombres entiers constitue un outil pour traiter les problèmes d'optimisation combinatoire.

L'objectif de cette thèse est d'étudier et d'utiliser les techniques d'optimisation combinatoire pour résoudre les problèmes de tomographie.

Mots-Clés : Tomographie discrète, Reconstruction d'images, Optimisation Combinatoire, Programmation mathématique, Programmation linéaire, Linéarisation et Convexification, Métaheuristiques.

Abstract

The tomographic imaging problem deals with reconstructing an object from a data called a projections and collected by illuminating the object from many different directions. A projection means the information derived from the transmitted energies, when an object is illuminated from a particular angle. The solution to the problem of how to reconstruct an object from its projections dates to 1917 by Radon. The tomographic reconstructing is applicable in many interesting contexts such as nondestructive testing, image processing, electron microscopy, data security, industrial tomography and material sciences.

Discrete tomography (DT) deals with the reconstruction of discrete object from limited number of projections. The projections are the sums along few angles of the object to be reconstruct. One of the main problems in DT is the reconstruction of binary matrices from two projections. In general, the reconstruction of binary matrices from a small number of projections is undetermined and the number of solutions can be very large. Moreover, the projections data and the prior knowledge about the object to reconstruct are not sufficient to determine a unique solution. So DT is usually reduced to an optimization problem to select the best solution in a certain sense.

In this thesis, we deal with the tomographic reconstruction of binary and colored images. In particular, research objectives are to derive the combinatorial optimization techniques in discrete tomography problems.

Keywords : Discrete Tomography, Reconstructing Image, Combinatorial Optimization, Integer programming, Convexification, Linearization.