

# Algorithme de coupe énumérative pour la minimisation de fonction d'énergie

## Cutting Enumerative Algorithm for the Minimizing of Energy Function

par M. DOUIMI\* et H. CHERIFI\*\*

\* IMHNSA de Rouen, URA CNRS 1378  
B.P. 08, Place Emile Blondel, 76 131 Mont-Saint-Aignan Cedex,  
Email : douimi@lmi.insa-rouen.fr  
\*\* TSI, UMR CNRS 842  
B.P. 505, 3 rue Javelin Pagnon, 42 007 Saint-Etienne

### *résumé et mots clés*

Nous étudions le problème de reconstruction de données unidimensionnelle par minimisation de la fonction d'énergie ou hamiltonien. Plusieurs méthodes de relaxation, stochastique et déterministe, ont été proposées pour résoudre ce problème d'optimisation globale et non convexe, notamment le recuit simulé (SA), le recuit par champ moyen (MFA), et la non convexité graduelle (GNC). Nous proposons un nouvel algorithme déterministe de coupe énumérative en optimisation combinatoire. Cet algorithme que nous appelons *Balayage par Position de Discontinuité (BPD)* permet d'obtenir des gains significatifs en terme de temps de calcul et de qualité de la solution obtenue. De plus, il est de nature parallélisable.

Optimisation globale et non convexe, champ de Markov, processus booléen, processus scalaire, algorithme GNC, hypercube, code de Gray

### *abstract and key words*

We study the 1-D reconstruction problem of real-valued data. This problem can be formulated by minimizing the energy function or hamiltonien. To solve this problem, many methods have been proposed such as, simulated annealing (SA), the graduated nonconvexity (GNC), and mean field annealing (MFA). We propose a new algorithm called "Discontinuity Position Sweep" (in French *Balayage par Position de Discontinuité : (BPD)*). This new algorithm comes from cutting enumerative methods in combinatorial optimization and its main advantages over the aforementioned methods are that it is parallelisable together, it is faster and it finds the better solution.

Non convex global optimization, Markov random field, ligne process, intensity process, GNC algorithm, hypercube, Gray code.

## 1. introduction

Au cours des dix dernières années, le traitement informatique des signaux (monodimensionnel et bidimensionnel) a émergé comme discipline majeure, tout en restant dépendante des branches traditionnelles de la physique et des mathématiques appliquées. Ce traitement est utilisé dans des domaines variés et son champ d'applications ne cesse de grandir. Différents objectifs apparaissent alors, se chevauchant et pouvant être traités suivant des méthodes

issues de différentes modélisations ou techniques. Ces méthodes plus ou moins adaptées dans chaque approche sont sensiblement différentes selon que l'on se place d'un point de vue théorique ou algorithmique. Il est intéressant de remarquer, que pour des signaux complexes, l'application d'une seule technique est parfois insuffisante. La solution pourrait être apportée par la coopération de méthodes [23].

Dans le cas du signal monodimensionnel, un des objectifs du traitement du signal est d'améliorer la qualité des signaux dans le but d'extraire plus facilement certains éléments comme, par exem-

ple, des changements brusques même si ces derniers font l'objet d'une étude directe; les références [5,6] dressent un panorama assez complet des divers algorithmes existants et des modélisations associées. L'amélioration de la qualité des signaux, autrement dit la restauration, peut être réalisée grâce à la reconstruction qui consiste à retrouver le signal à partir d'une version dégradée de celui-ci. La reconstruction a été intensivement étudiée depuis de nombreuses années en raison de son importance pratique, en particulier dans les domaines du traitement d'images et de la vision par ordinateur. Elle a surtout fait l'objet d'un travail poussé en formalisation mathématique.

La reconstruction appartient à la classe des problèmes inverses mal-posés au sens de Hadamard [4]. Pour résoudre ces problèmes, on utilise des techniques de régularisation par exemple la régularisation dite de Tikhonov [25]. Cette régularisation pénalise toutefois les discontinuités. Le lissage des discontinuités, ainsi que l'apparition d'oscillation, connues sous le nom de phénomène de Gibbs ont rapidement limité l'efficacité de ces approches et suscité ces dernières années un grand intérêt pour le développement de nouvelles méthodes, principalement non linéaires, qui assurent au mieux la prise en compte des discontinuités. On considère dans cet article une approche qui favorise le concept de processus de ligne pour marquer et préserver les discontinuités [14,2].

Un signal monodimensionnel numérique constant par morceaux est modélisé par un processus aléatoire markovien  $x$  (processus-scalaire). Le signal observé  $y$  est  $y = x + b$  où  $b$  est un bruit additif supposé blanc, gaussien centré, indépendant de  $x$  et d'écart-type  $\sigma$ . On introduit un processus  $l$  booléen modélisant les ruptures. Sous ces conditions, le couple  $(x, l)$  est estimé au sens du maximum a posteriori (MAP), c'est-à-dire en maximisant la probabilité conditionnelle aux observations  $y : P(x, l/y)$ . En utilisant le théorème de Bayes, on montre que la recherche de l'estimée optimale au sens du MAP se ramène au problème de minimisation d'une fonction d'énergie [14] :

$$E(x, l) = \frac{1}{2\sigma^2} \|x - y\|_2^2 + U(x, l) \quad (1)$$

où  $U(x, l) = \sum_{c \in \mathcal{C}} V_c(x_c, l_c)$  est la fonction d'énergie ou hamiltonien du système,  $\mathcal{C}$  étant l'ensemble des cliques qui correspondent au système choisi de voisinages,  $V_c(\cdot)$  est un potentiel local défini sur la clique  $c$  qui modélise les contraintes imposées par le modèle sur le signal reconstruit. Cette décomposition n'est pas unique, d'où la multiplicité des familles numériques de potentiels possibles.

On peut aboutir au même problème à partir d'une approche déterministe. Dans ce cas on modélise l'observation comme une fonction certaine, continue par morceaux, et dégradée par un bruit blanc additif, gaussien centré. On cherche alors à optimiser la distance entre l'observation et le signal en minimisant une fonction d'énergie qui combine des critères d'approximation

de régularité et de minimalité des ruptures. Cette formulation est celle retenue par A. Blake et A. Zisserman [2]. Ils mettent en évidence l'équivalence entre la minimisation de la fonction d'énergie induite par cette méthodologie et celle reposant sur la modélisation markovienne. Bien que le critère proposé soit uniquement interprété en terme de régularisation déterministe, sa forme est identique à celle de la log-vraisemblance d'un processus markovien.

Le problème de minimisation de la fonction d'énergie posé par (1) est un problème d'optimisation global mixte, non convexe, réputé très difficile [17]. Plusieurs méthodes ont été proposées pour le résoudre, on cite par exemple le recuit par champ moyen (MFA) [13], la non convexité graduelle (GNC) [2, 3], la programmation dynamique (DP) [22] et le recuit simulé (SA) [14]. Ces trois dernières méthodes ont été comparées au niveau de la complexité du temps et de la qualité de la solution par Blake[3].

Cet article propose un nouvel algorithme appelé Balayage par Position de Discontinuité (BPD) pour résoudre le problème d'optimisation. En optimisation combinatoire ce nouvel algorithme peut être classé comme un algorithme de coupe énumérative. L'algorithme GNC, pris comme référence [3, 10, 18, 24], appartient à la classe des algorithmes de relaxation déterministe. En traitement du signal, BPD est un algorithme primal-dual (zone homogène-rupture), car il fait intervenir au cours du déroulement les deux variables : une variable booléenne qui caractérise les ruptures (processus-booléen) et une variable réelle qui caractérise la zone homogène (processus-scalaire). Par contre GNC est un algorithme primal, car il ne fait intervenir que la variable réelle; la variable booléenne étant représentée implicitement par une fonction d'interaction locale.

BPD se distingue des autres algorithmes cités ci-dessus par les points suivants :

- Le temps d'exécution est indépendant du niveau du bruit;
- l'algorithme BPD n'est pas itératif et il donne une solution exacte dans le sens où elle est indépendante du choix du test d'arrêt;
- il est de nature parallélisable;
- on peut obtenir une solution peu coûteuse en temps de calcul pour des signaux de faible dimension.

L'organisation de l'article est la suivante. Dans la deuxième section nous formulons mathématiquement le problème et nous présentons un modèle pour le résoudre qui conduit à un problème d'optimisation. Dans la troisième section nous présentons et commentons brièvement l'algorithme GNC afin de mettre en évidence les différences essentielles avec l'algorithme BPD. Dans la quatrième section nous décrivons en détail l'algorithme du balayage par position de discontinuité. Enfin, dans la cinquième section, nous comparons le comportement expérimental des algorithmes GNC et BPD. Nous donnons quelques exemples de résultats sur des signaux synthétiques.

## 2. présentation d'un modèle

### 2.1. position du problème

Considérons le signal observé défini par :

$$y(t) = x(t) + b(t), \quad t \in [0, n] \quad (2)$$

où  $b(t)$  est un bruit gaussien centré d'écart-type  $\sigma$  et  $x(t)$  est un signal déterministe qui admet  $p$  points de discontinuités :

$$x(t) = \sum_{i=0}^p c_i \mathbb{1}_{[a_i, a_{i+1}[}(t), \quad t \in [0, n] \quad (3)$$

où  $\mathbb{1}_{[r, s[}(t) = \begin{cases} 1 & \text{si } t \in [r, s[ \\ 0 & \text{sinon} \end{cases}$  est la fonction caractéristique définie sur  $[r, s[$  avec  $a_0 = 0$ ,  $a_{p+1} = n$  et  $c_i \neq c_{i+1}$  pour  $i \in \{0, 1, \dots, p-1\}$ .

La reconstruction avec détection des points de discontinuité se ramène à la minimisation de la fonction d'énergie de référence  $E$ . Nous considérons une fonction d'énergie commune aux deux modélisations stochastique [14] et déterministe [2]. Elle est formée de la somme de trois éléments :

$$E = E_D + E_S + E_P \quad (4)$$

avec

\*  $E_D = \sum (x_i - y_i)^2$  : erreur quadratique moyenne assurant une bonne adéquation entre estimées et observées;

\*  $E_S = \lambda^2 \sum (x_i - x_{i+1})^2 (1 - l_i)$  : contrainte de lissage classique en fonction du processus-booléen  $l$ . Ce terme permet de filtrer les zones homogènes tout en conservant les discontinuités,  $\lambda$  étant un paramètre de lissage;

\*  $E_P = \alpha \sum l_i$  : coût à payer pour l'introduction des discontinuités,  $\alpha > 0$  étant un paramètre de pénalisation à l'introduction d'une discontinuité.

$x = (x_i)$  représente le signal reconstruit.  $l = (l_i)$  est un vecteur booléen introduit par Geman & Geman pour modéliser explicitement les ruptures [14] dans le cadre d'une approche markovienne. Ce processus est représenté par le diagramme suivant :

$$x_0 \xrightarrow{l_0} x_1 \xrightarrow{l_1} x_2 \dots x_{n-1} \xrightarrow{l_{n-1}} x_n$$

La détection d'un point de discontinuité à partir du processus intensité  $x = (x_i)$  est décidée de la façon suivante :

$$l_i = \begin{cases} 1 & \text{si } |x_i - x_{i+1}| > s \implies \text{discontinuité} \\ 0 & \text{sinon} \implies \text{continuité} \end{cases} \quad (5)$$

où  $s = \sqrt{\alpha}/\lambda$  est le seuil de détection des ruptures.

### 2.2. problème d'optimisation

Le signal reconstruit  $x$ , de positions de ruptures représentées par  $l$ , est un minimum global du problème d'optimisation à variable mixte et non convexe suivant :

$$(P) \quad \begin{cases} \text{minimiser} & E(x, l) \\ x \in \mathbb{R}^{n+1} \\ l \in \{0, 1\}^n \end{cases} \quad (6)$$

L'introduction du processus booléen conduit à minimiser une fonction non convexe (fonction quadratique par morceaux) qui admet plusieurs minimums locaux. Dans ces conditions, les méthodes classiques sont inopérantes [16] pour déterminer un minimum global.

Plusieurs méthodes ont été proposées pour résoudre ce problème d'optimisation, en 1989 Blake [3] a fait une étude comparative entre la qualité de la solution et la complexité du temps des différents algorithmes existant. Des algorithmes stochastiques, différentes variantes du recuit simulé [20, 14, 7] qui assurent la convergence asymptotique vers un minimum global au prix d'un temps d'exécution très élevé. Des algorithmes déterministes, le « Graduated NonConvexity » (GNC) proposé par Blake et Zisserman [2] et des algorithmes utilisant la technique de la programmation dynamique [22] dont une nouvelle variante est proposée par Blake [3]. Le GNC permet d'obtenir généralement une solution correcte (minimum global) en un temps d'exécution optimal. La technique de la programmation dynamique est utilisée comme un "assay" pour vérifier si la reconstruction obtenue est optimale. Son avantage est qu'elle fournit un minimum global; son inconvénient est la complexité en temps qui est de l'ordre de  $O(N^3)$  dans la pire des situations et de l'ordre de  $O(N^2)$  dans la meilleure des situations alors que la complexité de GNC est de l'ordre de  $O(\lambda N)$  avec  $\lambda \ll N$ .

Après 1989 d'autres algorithmes ont été proposés, on cite par exemple le recuit par champ moyen MFA [13], il est souvent affirmé que le MFA est légèrement inférieur à l'algorithme GNC, surtout, du point de vue de la qualité de la solution [18]. D'autres variantes de la programmation dynamique [1] sont proposées pour ne citer que la plus récente (1996) proposée par Alain de Cesare [8] qui réduit la complexité en temps à l'ordre de  $O(N)$ .

Il est à noter que le problème (6) utilise deux paramètres  $\lambda$  et  $\alpha$ . La qualité de la solution est fortement influencée par les valeurs de ces paramètres [2, 9]. Ceci implique qu'il faut sélectionner interactivement les paramètres (traitement par information ou calibration des paramètres). Dans la pratique on peut être amené à tester différents jeux de paramètres avant d'obtenir une solution satisfaisante.

## 3. algorithme GNC

### 3.1. élimination du processus booléen

Le problème d'optimisation (6) est un problème d'optimisation mixte puisqu'il fait intervenir deux variables de nature différente : une variable réelle et une variable booléenne. Il peut aussi être

## Algorithme de coupe énumérative

considéré comme un faux problème d'optimisation mixte car l'élimination de la variable booléenne est assez facile.

En effet : posons  $F(x) = \min_{l \in \{0,1\}^{n-1}} E(x, l)$ . Le calcul explicite de l'expression de cette nouvelle fonction objectif donne :

$$F(x) = E_D + \sum_{i=0}^n g_{\alpha, \lambda}(x_i - x_{i+1}) \quad (7)$$

$$\text{avec } g_{\alpha, \lambda}(t) = \begin{cases} \lambda^2 t^2 & \text{si } |t| < \sqrt{\alpha}/\lambda \\ \alpha & \text{sinon} \end{cases}$$

Ainsi le problème d'optimisation (6) est équivalent au problème suivant :

$$(P') \quad \begin{cases} \text{minimiser} & F(x) \\ x \in \mathbb{R}^{n+1} \end{cases} \quad (8)$$

## 3.2. non convexité graduelle

L'idée de base de GNC est la suivante :

**Première étape :** On construit une approximation convexe [21]  $F^*$  de  $F$ ,

$$F^*(x) = E_D + \sum_{i=0}^n g_{\alpha, \lambda}^*(x_i - x_{i+1}) \quad (9)$$

$$\text{avec } g_{\alpha, \lambda}^*(t) = \begin{cases} \lambda^2 t^2 & |t| < q \\ \alpha - 0.5c^* (|t| - r)^2 & q \leq |t| < r \\ \alpha & r \leq |t| \end{cases}$$

où  $c^* = 0.5$ ,  $r = \sqrt{\alpha(2/c^* + 1/\lambda^2)}$  et  $q = \alpha/(\lambda^2 r)$ .

La fonction  $F^*$  est de classe  $C^1$  partout, de classe  $C^2$  presque partout et strictement convexe (la situation où  $F^*$  est seulement convexe, est pratiquement irréalisable), par conséquent  $F^*$  admet un minimum local unique qui est donc global. Le calcul de ce minimum se fera alors avec des méthodes convenables de l'optimisation convexe<sup>1</sup> [19, 12].

**Deuxième étape :** Elle consiste à générer une séquence d'énergie  $(F^{(p)})_{p \in [0,1]}$ ,

$$F^{(p)}(x) = E_D + \sum_{i=0}^{n-1} g_{\alpha, \lambda}^{(p)}(x_i - x_{i+1}) \quad (10)$$

avec

$$g_{\alpha, \lambda}^{(p)}(t) = \begin{cases} \lambda^2 t^2 & |t| < q_p \\ \alpha - 0.5c_p (|t| - r_p)^2 & q_p \leq |t| < r_p \\ \alpha & r_p \leq |t| \end{cases}$$

où  $c_p = c^*/p$ ,  $r_p = \sqrt{\alpha(2/c_p + 1/\lambda^2)}$  et  $q_p = \alpha/(\lambda^2 r_p)$ .

La suite  $(F^{(p)})_{p \in [0,1]}$  évolue d'une manière continue en s'intercalant entre  $F^* = F^{(1)}$  approximation convexe et  $F = F^{(0)}$  la fonction objectif (7). Pour des raisons d'implémentation il est nécessaire de discrétiser cette suite, autrement dit de discrétiser l'intervalle  $]0, 1[$ . Il est prouvé [2] que la discrétisation suivante  $\Delta = \left\{ \dots, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1 \right\}$  de  $]0, 1[$  est convenable dans

1. Des méthodes du premier et second ordre sont proposées dans [12].

le sens où elle fournit généralement une solution correcte en un temps d'exécution optimal. L'algorithme GNC optimise toute la séquence d'énergie discrète  $(F^{(p)})_{p \in \Delta}$ . Comme  $F^{(p)}$ , pour  $p \neq 1$ , admet plusieurs minimums locaux, le choix du point initial est très important dans ce cas. La stratégie utilisée consiste à choisir la solution trouvée lors de la minimisation de la fonction précédente comme condition initiale pour déterminer le minimum global de la fonction courante. A priori, cette stratégie augmente la probabilité de déterminer le vrai minimum global de la fonction courante. En effet, le point du départ par construction (voir figure (1)) est voisin de ce minimum puisque la fonction courante et la fonction précédente sont voisines. On note que seule la première fonction  $F^{(1)} = F^*$  échappe à cette règle grâce à la convexité (voir étape 1).

## 3.3. algorithme GNC

La suite de fonction  $(F^{(p)})_{p \in \Delta}$  est continue et à dérivées premières continues. Pratiquement, toutes les méthodes d'optimisations sans contraintes dans  $\mathbb{R}^n$  consistent à rechercher un point  $x^*$  caractérisé par la condition de stationnarité

$$\nabla F^{(p)}(x^*) = 0. \quad (11)$$

Dans le cas des méthodes de gradient, il s'agit de procédures itératives. On engendre une suite de points  $x^{(0)}, x^{(1)}, x^{(2)}, \dots, x^{(k)}$  convergeant vers un point optimum local de  $F^{(p)}$ . à chaque étape  $k$ ,  $x^{(k+1)}$  est définie par :

$$x^{(k+1)} = x^{(k)} - w P^{-1} \nabla F(x^{(k)}) \quad (12)$$

où  $w$  est le pas de déplacement, calculé en utilisant une approximation quadratique locale qui n'est autre que le paramètre de relaxation optimale (SOR). La direction de déplacement  $-\nabla F(x^{(k)})$  est préconditionnée par la matrice

$$P = \text{diag} \left( \left| \frac{\partial^2 F}{\partial x_i \partial x_j} \right| \right).$$

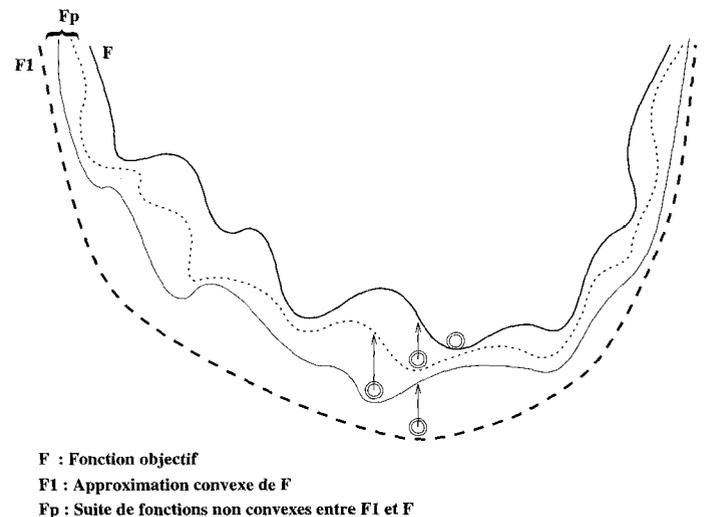


Figure 1. – Schéma explicatif du principe de l'algorithme de GNC.

Choisir  $\lambda, h$

Calculer  $\alpha = h^2\lambda/2$

Paramètre : SOR  $w = 2\lambda/(1 + \lambda)$

La suite de fonctions :  $p \in \{1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{\lambda}, \frac{1}{2\lambda}\}$

Sites :  $i \in \{0, \dots, n - 1\}$

Pour tout  $p$ , itérer  $k = 1, 2, \dots$

Pour  $i = 1, \dots, n - 2$

$$x_i^{(k+1)} = x_i^{(k)} - 2T(x_i^{(k)} - y_i) + T\{g_{\alpha,\lambda}^{(p)'}(x_i^{(k)} - x_{i-1}^{(k+1)}) + g_{\alpha,\lambda}^{(p)'}(x_i^{(k)} - x_{i+1}^{(k+1)})\}$$

Les modifications apportées aux bords :

$$x_0^{(k+1)} = x_0^{(k)} - 2T_0(x_0^{(k)} - y_0) + T_0g_{\alpha,\lambda}^{(p)'}(x_0^{(k)} - x_1^{(k)})$$

$$x_{n-1}^{(k+1)} = x_{n-1}^{(k)} - 2T_0(x_{n-1}^{(k)} - y_{n-1}) + T_0g_{\alpha,\lambda}^{(p)'}(x_{n-1}^{(k)} - x_{n-2}^{(k)})$$

avec  $T = w/(2 + 4\lambda^2), T_0 = w/(2 + 2\lambda^2)$  et

$$g_{\alpha,\lambda}^{(p)'}(t) = \begin{cases} 2\lambda^2 t & \text{si } |t| < q_p \\ -\frac{t}{2p|t|}(|t| - r_p) & \text{si } q_p \leq |t| < r_p \\ 0 & \text{si } r_p \leq |t| \end{cases}$$

où

$$r_p = \sqrt{\alpha(4p + 1/\lambda^2)} \quad \text{et} \quad q_p = \alpha/(\lambda^2 r_p)$$

L'algorithme *GNC* dans le cas d'un signal monodimensionnel est le suivant : L'algorithme *GNC* (*GNC-SOR*) présenté ci-dessus pour le cas d'un signal monodimensionnel est un algorithme itératif. Il fournit un  $\varepsilon$ -solution dans le sens où la solution optimale dépend du choix du test d'arrêt, par exemple ( $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$ ).

### 3.4. convergence de GNC

La convergence de l'algorithme *GNC* repose sur le principe de transfert par continuité des minimums globaux des fonctions  $F^{(1)}, F^{(\frac{1}{2})}, \dots, F^{(\frac{1}{2\lambda})}$ . Cela signifie que si l'on sait calculer un minimum global  $x_p^*$  de  $F^{(p)}$  alors, on peut atteindre un minimum global  $x_p^*$  de  $F^{(p)}$  ( $p' < p$ ) à l'aide d'un algorithme de type local avec le point initial  $x_1^*$ . Puisque  $F^{(1)}$  est strictement convexe, on peut calculer son minimum global  $x_1^*$  qui servira à initialiser un algorithme local dans le calcul d'un minimum local de  $F^{(p)}$  avec  $p \in ]0, 1[$  suffisamment proche de 1, et ainsi de suite, comme dans un processus dynamique, jusqu'à l'obtention d'un minimum global de  $F$ . On ne dispose pas à l'heure actuelle d'une étude théorique rigoureuse de la convergence de l'algorithme *GNC*. Il est aussi important d'ajouter que la convergence de *GNC* est sensible au choix du test d'arrêt qui influ sur la précision des solutions globales  $x_p^*$  de  $F^{(p)}$ . Par conséquent, *GNC* pourrait ne pas converger vers un minimum global de  $F$  s'il ne converge pas vers un des minimums globaux  $x_p^*$  (voir figure (2)) de la

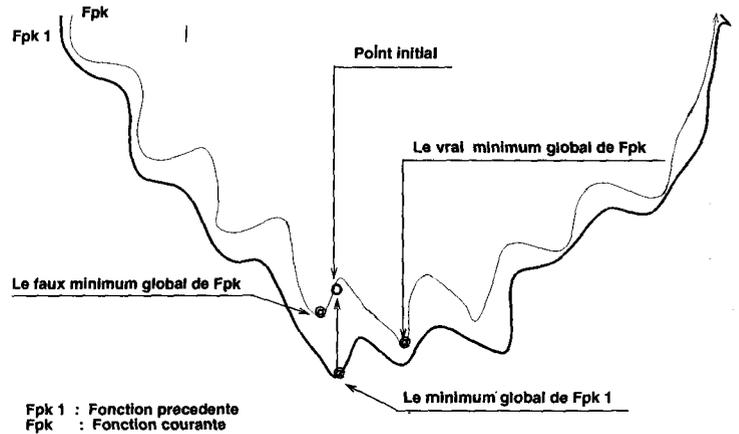


Figure 2. - Schéma explicatif du cas où on peut ne pas converger vers le minimum global de  $F_{pk-1}$ .

suite  $F^{(p)}$ . Cette situation est peu fréquente dans les applications pratiques. En général, l'algorithme *GNC* permet d'obtenir une bonne estimation d'un minimum global de  $F$ .

## 4. algorithme BPD

### 4.1. modélisation hypercubique

On utilise exclusivement dans la suite de cet article l'ensemble des processus-booléens  $\Omega = \{0, 1\}^n$  qui est modélisé par un graphe hypercube. Un hypercube de degré  $n$  (ou  $n$ -Cube) est un réseau à  $2^{n-1}$  nœuds, où chaque nœud a exactement  $n$  voisins; on trouvera un hypercube de degré 4 (4-Cube) à la figure (4). La construction d'un hypercube est récursive [11] : on déduit un  $n$ -Cube à partir de deux  $(n - 1)$ -Cube où chaque nœud de l'un est relié au nœud de même position relative de l'autre (l'hypercube de degré 0 étant un simple nœud isolé); ceci est illustré par la figure (3).

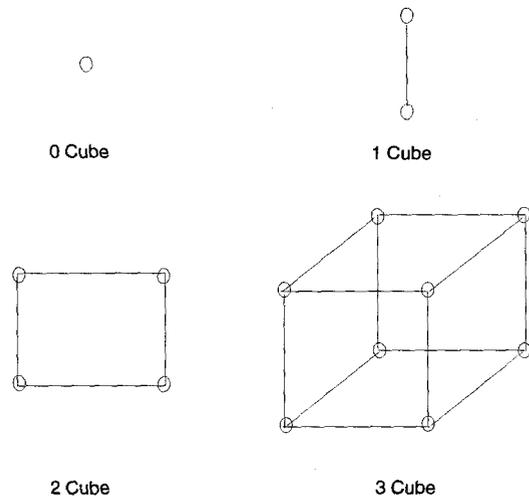


Figure 3. - Construction récursive d'un hypercube.

## Algorithme de coupe énumérative

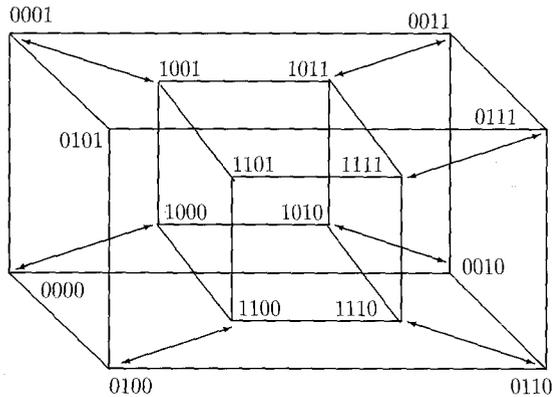


Figure 4. – Hypercube de degré 4.

Il existe une numérotation commode à partir des codes de Gray où chaque nœud du réseau est un entier codé sur  $n$  bits en binaire (vecteur booléen de composante 0 ou 1), dont on déduit tous les voisins en complétant successivement chaque bit (composante) de cette numérotation; par conséquent, chaque nœud a exactement  $n$  voisins.

Les nœuds de l'hypercube  $\Omega = \{0, 1\}^n$  énumèrent les processus booléens que l'on indexe par les vecteurs booléens

$\{e^I\}_{I \subset \{0, 1, \dots, n\}}$  où la  $k^{\text{ième}}$  composante de  $e^I$  est définie par :

$$e_k^I = \begin{cases} 1 & \text{si } k \in I \\ 0 & \text{sinon} \end{cases} \quad (13)$$

### 4.2. énumération des minimums locaux

Notons  $E_I(x) = E(x, e^I)$  la fonction énergie à processus booléen  $e^I$  fixé. Cette fonction est quadratique strictement convexe donc admet un minimum local  $x^I$  unique, caractérisé par la condition de stationnarité

$$\nabla E_I(x^I) = 0 \quad (14)$$

Pour déterminer le minimum global  $x^I$  de  $E_I$ , il suffit alors d'expliciter (14) ce qui revient à résoudre un système linéaire

$$A(I)x^I = y \quad (15)$$

où la matrice  $A(I)$  est tridiagonale symétrique et définie positive [15], matrice que nous écrivons de manière condensée  $A = (a, b, c)$  avec :

$$\begin{aligned} a &= (0, -\lambda^2(1 - e_0^I), \dots, -\lambda^2(1 - e_{n-1}^I)) \\ b &= (1 + \lambda^2(1 - e_0^I), 1 + \lambda^2(2 - e_0^I - e_1^I), \dots, \\ &\quad 1 + \lambda^2(2 - e_{n-2}^I - e_{n-1}^I), 1 + \lambda^2(1 - e_{n-1}^I)) \\ c &= (-\lambda^2(1 - e_0^I), \dots, -\lambda^2(1 - e_{n-1}^I), 0) \end{aligned}$$

La matrice  $A(I)$  est de la forme :

$$A(I) = \begin{bmatrix} b_0 & c_0 & 0 & \dots & 0 \\ a_1 & b_1 & \dots & \dots & \vdots \\ 0 & \dots & \dots & \dots & 0 \\ \vdots & \dots & \dots & b_{n-1} & c_{n-1} \\ 0 & \dots & 0 & a_n & b_n \end{bmatrix} \quad (16)$$

Les points stationnaires susceptibles d'être des minimums locaux de  $E(x, e^I)$  sont caractérisés par l'équation (15) d'où le problème (P) est équivalent au problème

$$(P'') \quad \begin{cases} \text{minimiser} & E(x, e^I) \\ A(I)x = y \\ x \in \mathbb{R}^{n+1} \\ e^I \in \Omega \end{cases} \quad (17)$$

que l'on simplifie de la manière suivante :

$$(P''') \quad \begin{cases} \text{minimiser} & E(A(I)^{-1}y, e^I) \\ e^I \in \Omega \end{cases} \quad (18)$$

La résolution de ce dernier problème d'optimisation combinatoire est théoriquement facile mais pratiquement très coûteuse (voire impossible) vu que  $\text{card}(\Omega) = 2^{n+1}$ . On détermine le minimum global qui sera dans ce cas la plus petite évaluation énergétique de tous les points stationnaires susceptibles d'être des minimums locaux (solution des systèmes d'équations linéaires (15)). Cette stratégie consiste à explorer les  $2^{n+1}$  configurations, ce qui peut être très coûteux en temps de calcul. Nous proposons, alors, une stratégie de coupe qui réduit le nombre de configurations à examiner.

### 4.3. stratégie de coupe

Décomposons  $\Omega$  en plusieurs sous-ensembles disjoints

$$\Omega = \bigsqcup_{k=0}^n \Omega_k \quad (19)$$

Ces ensembles  $\Omega_k = \{e^I \in \{0, 1\}^n / \text{card}(I) = k\}$  sont les  $(n+1)$  niveaux de l'hypercube  $\Omega$ . Le niveau  $\Omega_k$  de l'hypercube  $\Omega$  rassemble tous les processus booléens qui admettent  $k$  positions de discontinuités. Autrement dit  $\Omega_k$  est l'ensemble des processus booléens qui représentent toutes les configurations possibles de  $k$  positions de ruptures du signal à reconstruire. On construit alors la suite de problèmes d'optimisation indépendants correspondant à cette décomposition :

$$(P_k) \quad \begin{cases} \text{minimiser} & E(A(I)^{-1}y, e^I) \\ e^I \in \Omega_k \end{cases} \quad (20)$$

avec  $k = 0, \dots, n$ .

La solution optimale du problème  $(P_k)$  correspond au signal reconstruit ayant pour processus booléen  $k$  composantes égales à 1 ( $k$  points de discontinuité). Par exemple, la résolution du

problème ( $P_0$ ) se réduit à celle du système linéaire  $A(\phi)x = y$ ; la solution obtenue correspond à un signal reconstruit continu. La résolution des problèmes ( $P_k$ ) est facile mais coûteuse, sauf pour ( $P_0$ ) et ( $P_n$ ) puisqu'elle consiste à résoudre  $\binom{n}{k}$  systèmes d'équations linéaires.

L'idée de base est de réduire le coût de la résolution de chaque problème ( $P_k$ ), en réduisant le domaine d'optimisation  $\Omega_k$  en un sous-ensemble de celui-ci, l'ensemble  $\mathcal{V}_s(e^{I_{k-1}})$  des voisins du niveau supérieur du nœud  $e^{I_{k-1}}$

$$\mathcal{V}_s(e^{I_{k-1}}) = \bigcup_{i \in I_{k-1}^c} \{e^{I_{k-1} \cup \{i\}}\} \quad (21)$$

avec  $I_{k-1}^c = \{1, \dots, n\} \setminus I_{k-1}$ . On a donc  $\mathcal{V}_s(e^{I_{k-1}}) \subset \Omega_k$  et  $\text{card}(\mathcal{V}_s(e^{I_{k-1}})) = n - \text{card}(I_{k-1}) - 1$ .

Transformons la suite de problèmes d'optimisation ( $P_k$ ) en une suite de problèmes ( $Q_k$ ) récurrente et beaucoup moins coûteuse en résolution de système d'équations linéaires.

$$(Q_k) \quad \begin{cases} \text{minimiser} & E(A(I)^{-1}y, e^I) \\ e^I \in \mathcal{V}_s(e^{I_{k-1}}) \end{cases} \quad (22)$$

avec  $k = 1, \dots, n$  où  $I_{k-1}$  est l'ensemble des positions de discontinuité du signal reconstruit solution optimale du problème ( $Q_{k-1}$ ) avec  $I_0 = \phi$ .

La suite de problèmes d'optimisation ( $Q_k$ ) est récurrente car pour résoudre le problème ( $Q_k$ ) nous exploitons l'information apportée par la solution optimale du problème ( $Q_{k-1}$ ), à savoir  $k - 1$  positions de ruptures du signal reconstruit. C'est cette information qui nous a permis de faire une sorte de restriction du domaine d'optimisation du processus booléen, domaine égal à  $\Omega_k$  pour le problème ( $P_k$ ) et à  $\mathcal{V}_s(e^{I_{k-1}})$  pour le problème ( $Q_k$ ).

La résolution des problèmes ( $Q_k$ ) est facile et moins coûteuse que celle des ( $P_k$ ) puisqu'elle consiste pour chaque  $k$  fixé à résoudre  $(n - k)$  systèmes d'équations linéaires au lieu de résoudre  $\binom{n}{k}$  systèmes d'équations linéaires pour le problème ( $P_k$ ).

#### 4.4. algorithme BPD

Nous avons appelé ce nouvel algorithme *balayage par position de discontinuité* car pour la résolution de chaque problème ( $Q_k$ ) nous balayons les positions de continuités (composante binaire égale à 0) du processus booléen  $e^{I_{k-1}}$  avec  $(x^{I_{k-1}}, e^{I_{k-1}})$  solution du problème ( $Q_{k-1}$ ), par une position de discontinuité (composante binaire égale à 1). Cette opération est formulée mathématiquement par le choix du domaine d'optimisation de la variable processus-booléen. Chaque nouvelle position de discontinuité est retenue ou rejetée selon la comparaison de son énergie avec l'énergie du signal reconstruit sans cette nouvelle position de discontinuité détectée. Ce qui se traduit à la  $k^{\text{ième}}$  détection de rupture par :

\* Si  $E_{I_k} \leq E_{I_{k-1}}$  la nouvelle position  $i_k$  de discontinuité détectée est retenue comme étant la  $k^{\text{ième}}$  position de discontinuité du futur signal reconstruit (minimum global à déterminer).

\* Sinon cette nouvelle position  $i_k$  de discontinuité est rejetée et le signal reconstruit, recherché, ne peut admettre plus de  $(k - 1)$  positions de discontinuité. Nous proposons l'algorithme suivant :

---

##### étape 1

- Choisir  $\lambda$  et  $h$  (lissage et sensibilité)
- $k = 0$
- $I_k = \phi$
- $E_{I_k} = E(x^{I_k}, e^{I_k})$

##### étape 2

- $k = k + 1$
- Résoudre le problème :

$$(Q_k) \quad \begin{cases} \text{minimiser} & E(A(I)^{-1}y, e^I) \\ e^I \in \mathcal{V}_s(e^{I_{k-1}}) \end{cases}$$

- $i_k$  est la  $k^{\text{ième}}$  position de rupture détectée
- $(x^*, e^*)$  est la solution optimale de ( $Q_k$ )
- $E_{I_k} = E(x^*, e^*)$

- Si  $E_{I_k} \leq E_{I_{k-1}}$  faire

$$\begin{cases} * I_k = I_{k-1} \cup \{i_k\} \\ * \text{Aller à l'étape 2} \end{cases}$$

Sinon faire

$$\begin{cases} * (x^{I_{k-1}}, e^{I_{k-1}}) \text{ solution optimale} \\ * \text{Fin} \end{cases}$$


---

##### Remarques :

• L'algorithme BPD est de nature parallélisable car la résolution de l'ensemble des systèmes linéaires d'un sous-ensemble d'un niveau  $k$  de l'hypercube (résolution du problème d'optimisation ( $Q_k$ )) se fait d'une manière indépendante.

• La solution donnée par l'algorithme BPD est exacte, elle ne dépend pas du choix d'un test d'arrêt, contrairement à l'algorithme GNC.

• Si le nombre de discontinuités est connu c'est un plus de l'algorithme BPD par rapport aux autres algorithmes car cette information peut être utilisée comme un test d'arrêt intéressant pour cet algorithme ce qui n'est pas le cas de GNC.

#### 4.5. convergence de BPD

L'algorithme BPD sort de la classe des algorithmes classiques qui sont généralement itératifs et à convergence infinie; cela est dû à son caractère de coupe énumérative. Tout comme pour les méthodes de relaxation déterministe l'étude de la convergence de cet algorithme est très complexe. En effet, cela nécessite de calculer explicitement l'énergie ou la variation d'énergie des couples  $(A(I)^{-1}y, e^I)$  avec  $e^I \in \Omega$ ; autrement dit calculer explicitement la solution du système  $A(I)x = y$ . Néanmoins, dans l'hypothèse où l'observation est non bruitée, on montre dans [12] que la solution obtenue par BPD est un minimum global du problème d'optimisation posé. Pour une observation bruitée, on donne des éléments pour l'extension de ces résultats.

## 5. résultats expérimentaux

L'exemple suivant illustre bien le comportement de GNC et de BPD : Considérons le signal déterministe figure (5, (a)) de dimension  $n = 200$  comportant 4 points de rupture définis par (position, amplitude) : (40, 20), (60, -40), (100, 50) et (160, -30) noyé dans un bruit blanc gaussien centré d'écart type  $\sigma = 18$ , figure (5, (b)). Les différentes étapes de l'estimation du signal bruité de la figure (5) sont illustrés par la figure (6) par l'algorithme GNC et par la figure (7) par l'algorithme BPD.

Le rapport signal sur bruit (RSB) est défini, dans ces tests, comme suit :

$$RSB = 20 \text{Log}_{10} \left( \frac{H}{\sigma} \right) \text{ avec } H = \min_{i \in [1,p]} |c_i - c_{i-1}| \quad (23)$$

où  $H$  représente la valeur de la plus petite amplitude de discontinuité.

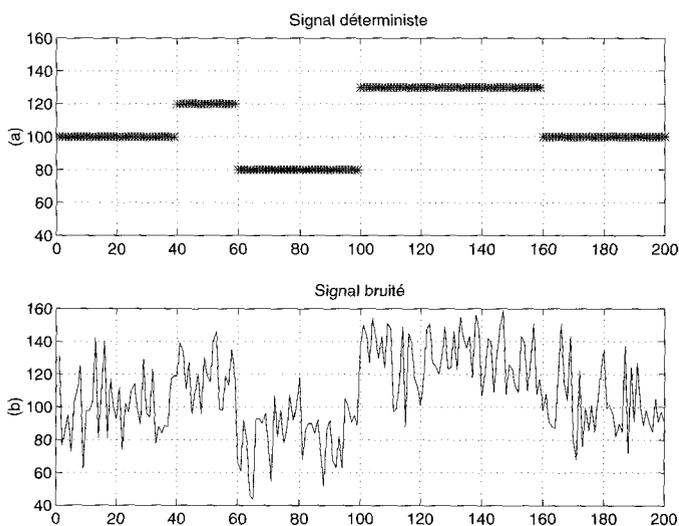


Figure 5. – (a) : Signal déterministe. (b) : Observation bruitée.

### 5.1. comportement de GNC

La figure (6) représente les minimums de la suite de fonctions  $\{F^{(p)}, \text{ avec } p = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$ . Ces minimums peuvent être schématisés grossièrement comme étant des organisations des points du signal  $(i, x(i))$  pour  $i = 0, \dots, n$ , en partant d'une organisation initiale qui est imposée par le signal bruité. Au fur et à mesure le paramètre  $p$  décroît; il y a une sorte de réorganisation des points du signal (minimum global de  $F^{(p)}$ ) qui tend vers une organisation stable. Celle-ci correspond théoriquement à la valeur  $1/(2\lambda)$  du paramètre  $p$ . Cette organisation finale n'est autre qu'un minimum global de la fonction énergie.

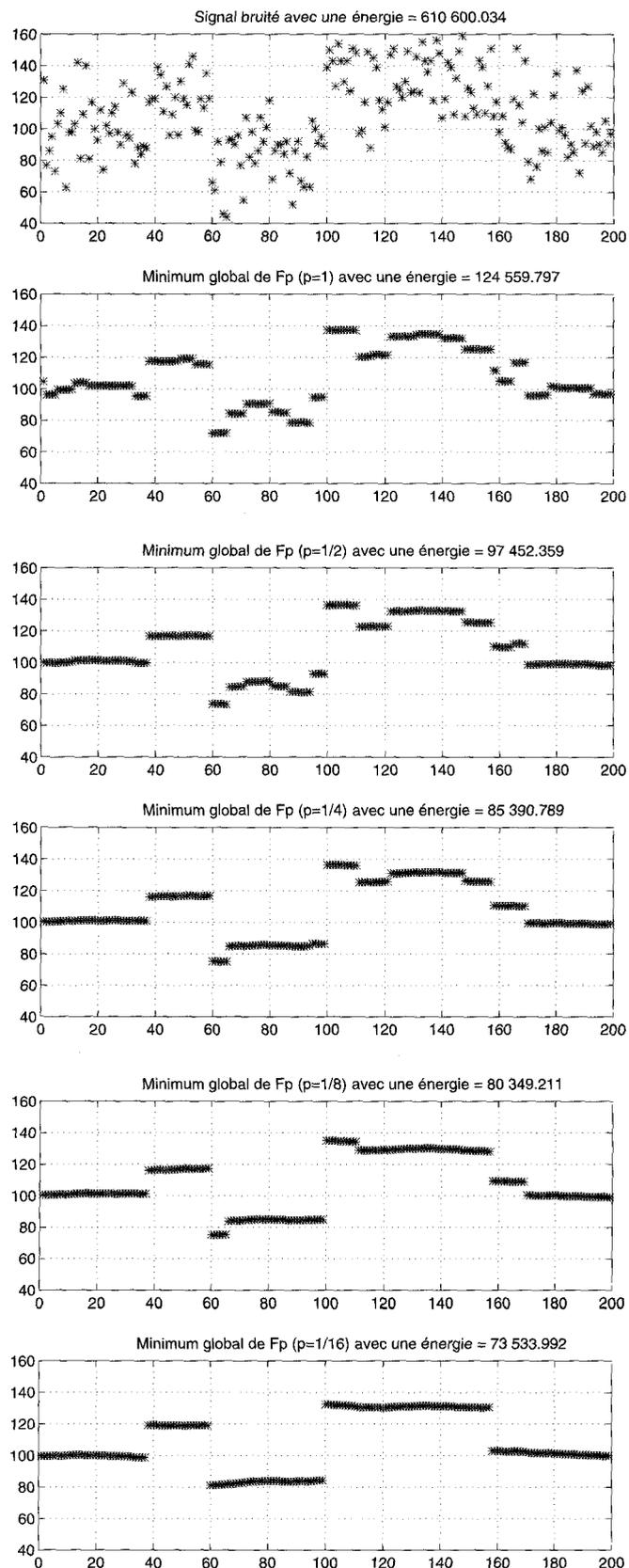


Figure 6 – Signal reconstruit pour les différents minimums de  $F^{(p)}$ .

### 5.2. comportement de BPD

Les figures (7 (a), (b), (c), (d) et (e)) représentent les différentes étapes de détection des ruptures. La position du minimum global de l'énergie du niveau 1 de l'hypercube figure (7 (a)), correspond à celle du point de rupture de la plus grande amplitude (position 100, amplitude de saut 52). Nous constatons que le point de discontinuité suivant est déjà repéré aux environs de la position 60 et sera confirmé par le passage au niveau 2 figure (7 (b)). Ce processus se répète pour les ruptures suivantes (figures (7 (c) et (d))) jusqu'à ce que l'énergie minimum du niveau courant (niveau 4 figure (7 (d)) dans notre exemple) soit supérieur à celui du niveau précédent (5) (figure (7 (e))). L'estimation des points de discontinuité est résumé dans le tableau 1. Nous constatons que la détection des ruptures se fait séquentiellement non pas en fonction des positions de ces dernières, mais plutôt en fonction de leur amplitude.

Tableau 1. - Positions et amplitudes des ruptures

Figure	(a)	(b)	(c)	(d)
Position	100(+0)	60(+0)	159(-1)	39(-1)
Amplitude	43	-31	-29	20
Energie	84 602	78 306	73 953	73 533

L'erreur d'estimation des positions des discontinuités est liée au fort niveau de bruit choisi dans cet exemple. Dans les mêmes conditions ( $\lambda = 20, h = 18$ ), on obtient une estimation identique du signal avec un traitement par GNC.

### 5.3. temps d'exécution de GNC

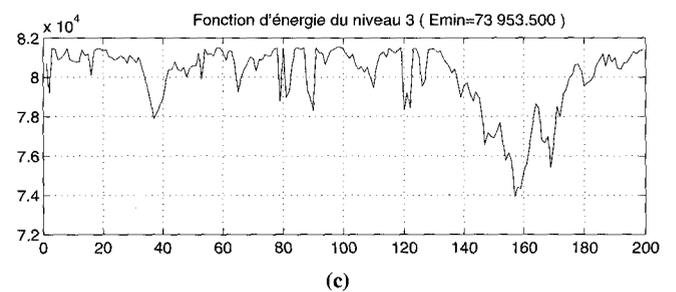
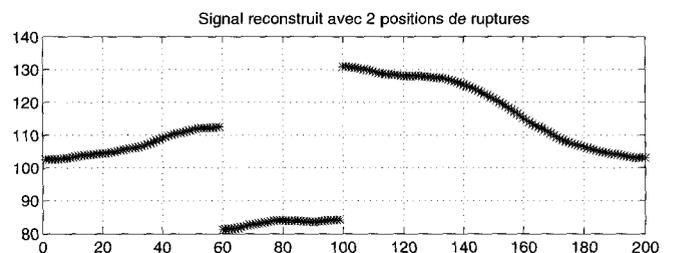
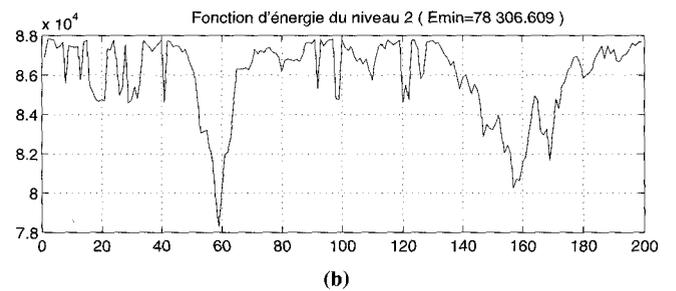
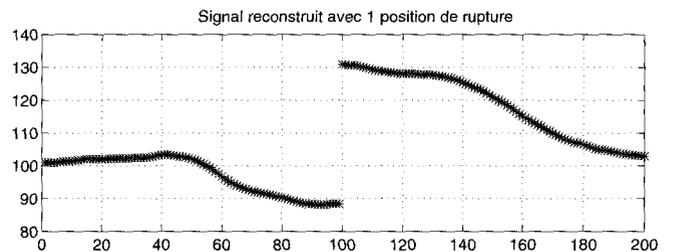
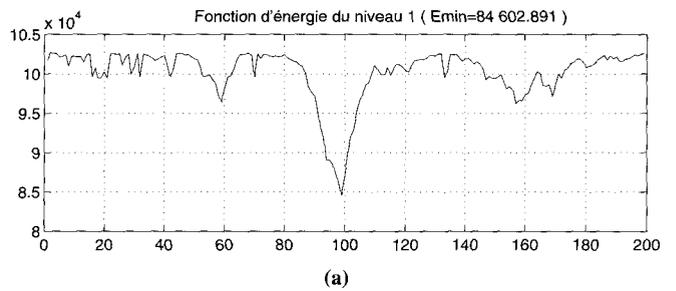
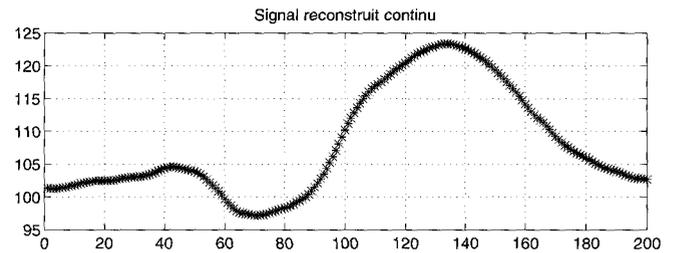
Le temps d'exécution de l'algorithme GNC dépend des facteurs suivants que nous classons par ordre d'importance

\* Le niveau du bruit : on remarque une nette augmentation (en moyenne) du temps d'exécution de l'algorithme GNC quand celui-ci devient important (voir tableau (2));

\* Le test d'arrêt : il influe sur le temps d'exécution et la qualité de la solution [12];

Tableau 2. - Le temps d'exécution de GNC en fonction de la dimension du signal et de l'écart-type du bruit avec une précision du test d'arrêt de 0.001.

Dimensions	écart-Type du bruit $\sigma$				
	4	8	12	16	20
100	1.03 s	2.75 s	1.67 s	4.18 s	4.53 s
150	1.09 s	1.68 s	2.63 s	6.83 s	11.92 s
200	1.93 s	2.92 s	7.65 s	6.12 s	16.27 s
250	1.67 s	4.40 s	8.32 s	12.42 s	23.23 s
300	3.58 s	9.17 s	8.33 s	11.18 s	30.45 s
350	2.32 s	4.12 s	15.62 s	20.42 s	31.82 s
400	3.57 s	10.17 s	7.27 s	24.07 s	31.28 s
450	2.97 s	10.55 s	11.55 s	32.07 s	59.18 s
500	3.48 s	8.00 s	17.83 s	21.62 s	55.66 s



## Algorithme de coupe énumérative

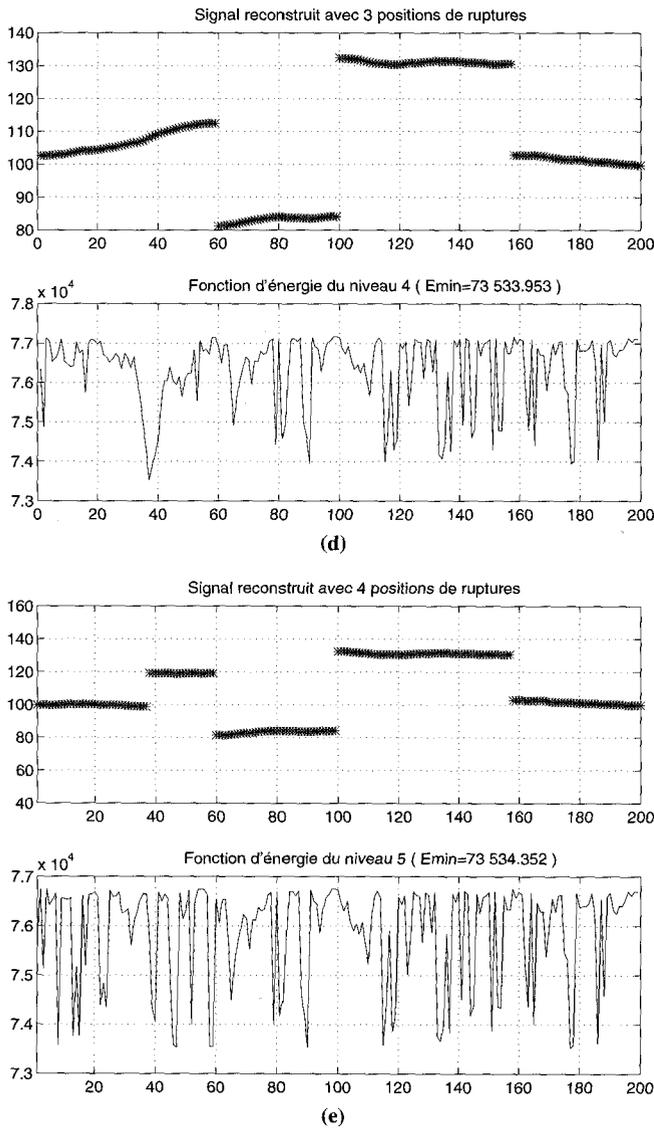


Figure 7. – Comportement de BPD.

\* La dimension du signal et le nombre de ruptures détectées, ces paramètres conditionnent le nombre d'opérations arithmétiques.

Les valeurs du tableau (2) représentent les temps d'exécutions de GNC pour l'estimation d'un signal ayant une seule position de rupture (se trouvant au milieu) de saut d'amplitude constant 20, de dimensions indiquées dans la première colonne du tableau, noyé dans un bruit gaussien, régénéré pour chaque test, ayant pour écart-type les valeurs de la première ligne du tableau.

Ces tests ont été effectués sur une station SPARC 10, le test d'arrêt de l'algorithme GNC étant de  $10^{-3}$ .

### 5.4. temps d'exécution de BPD

Le temps d'exécution de l'algorithme BPD ne dépend que de la dimension du signal et du nombre de ruptures détectées, figure (8).

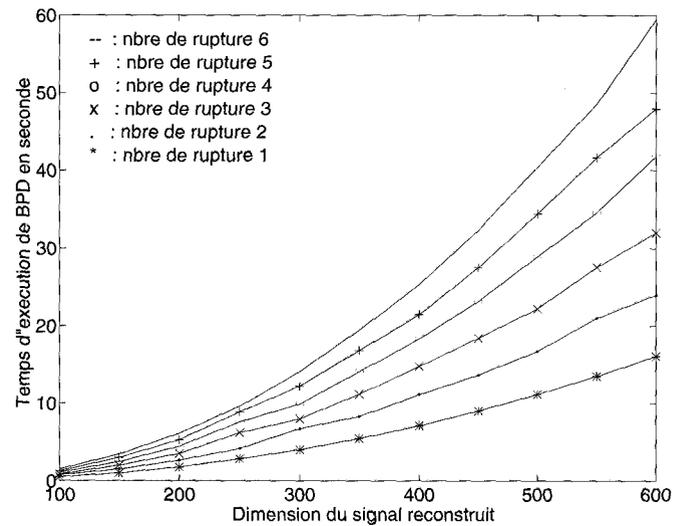


Figure 8. – Le temps d'exécution de l'algorithme BPD en fonction du nombre de ruptures et de la dimension du signal.

En effet le nombre de systèmes d'équations linéaires à résoudre croît avec ces deux facteurs. Par contre le temps d'exécution est indépendant du niveau de bruit. Ce qui destine tout naturellement cet algorithme pour des signaux de dimension faible fortement bruités.

### 5.5. quelques exemples illustratifs

Les algorithmes GNC et BPD ont été comparés sur des données simulées. Au cours des tests numériques, nous avons eu l'occasion de traiter quelques exemples intéressants illustrés par la figure (9) et le tableau (3).

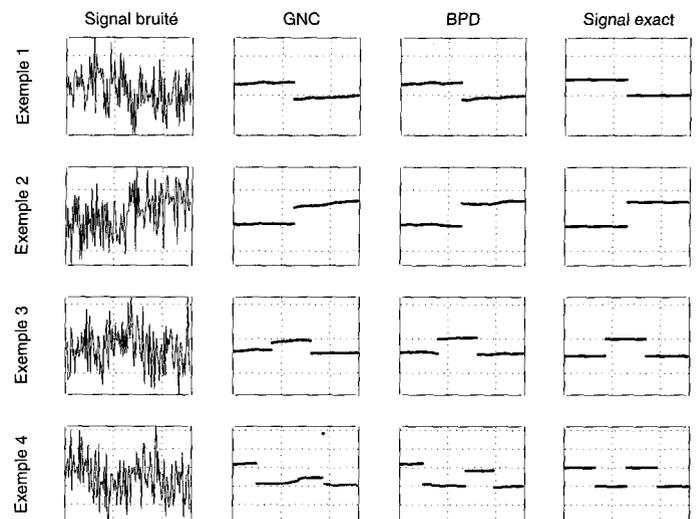


Figure 9. – Exemples de traitement.

Tableau 3. – Le temps d'exécution et l'énergie des exemples de la figure (9) par les algorithmes GNC et BPD

	temps (s)		énergies	
	GNC	BPD	GNC	BPD
(1)	6.30 s	0.40 s	65 458.316	65 337.273
(2)	2.70 s	0.42 s	33 984.586	33 675.023
(3)	6.00 s	0.57 s	105 778.609	104 084.344
(4)	7.67 s	0.77 s	52 281.059	52 076.738

• **Exemple 1 :** 1 rupture, SNR = -2.50 : pour  $\lambda = 20$  et  $h = 18$ , le temps d'exécution augmente avec le bruit pour GNC, BPD échappe à cette règle.

• **Exemple 2 :** 1 rupture, SNR = 1.94 : pour  $\lambda = 20$  et  $h = 18$ , le temps d'exécution de l'algorithme GNC est fortement lié au niveau du bruit, alors que ce n'est pas le cas pour BPD.

• **Exemple 3 :** 2 ruptures, SNR = 1.94 : pour  $\lambda = 25$  et  $h = 20$ , le GNC et le BPD détectent les mêmes positions de rupture avec un avantage au BPD, au niveau de l'évaluation de l'énergie des solutions. Cela se traduit par une meilleure reconstruction.

• **Exemple 4 :** 3 ruptures, SNR = 0.00 : pour  $\lambda = 20$  et  $h = 14$ , l'algorithme GNC n'atteint pas le minimum global (voir comparaison des énergies du tableau (3)), alors que l'algorithme BPD donne une meilleure solution.

Les différents résultats sont résumés dans le tableau (3).

## 6. conclusion et perspectives

Nous avons présenté un nouvel algorithme d'optimisation pour la reconstruction et la détection des positions de ruptures d'un signal dégradé par un bruit blanc, gaussien centré, additif. Cet algorithme permet de résoudre le problème de façon efficace pour des signaux de faible dimension comportant un nombre raisonnable de discontinuités. Il présente de nombreux avantages par rapport aux algorithmes de relaxation déterministe et stochastique. C'est un algorithme primal-dual, non itératif, et de nature parallélisable. La solution est exacte et le temps de calcul est indépendant du niveau du bruit.

Notre objectif final est d'adapter cet algorithme pour les signaux bidimensionnels. Pour ce faire deux voies complémentaires sont à explorer. La première consiste à accroître les performances de la version séquentielle et unidimensionnelle de l'algorithme BPD par l'intégration d'un certain nombre de propriétés déduites de l'expérimentation. La seconde voie consiste à exploiter la nature parallèle de cet algorithme.

### BIBLIOGRAPHIE

[1] A.A. Amiri, T.E. Weymouth & R.C. Jain (1990), «Using dynamic programming for solving variationnel problems en vision», *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol 12, n°, pp 855-867, Septembre 1990.

[2] A. Blake & A. Zisserman (1987), *Visual Reconstruction*, MIT Press, Cambridge 1987.

[3] A. Blake (1989), «Comparaison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction», *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 11, janv. 1989.

[4] M. Bertero, T.A. Poggio & V. Torre (1988), «Ill-posed problems in early vision», *Proceedings of the IEEE* Vol. 76, n° 8, pp. 869-889, Août 1988.

[5] M. Basseville & B. Espiau & J. Gasnier (1981), «Edge Detection using Sequential Methods for Change in Level - Part I : A sequential edge detection algorithm», *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSSP-29, n° 1, Feb. 1981, pp. 24-31.

[6] M. Basseville & A. Benhallam & all. (1992), «Fiches d'algorithmes de segmentation de signaux», *Traitement du signal*, Supplément au vol. 9, n° 1, 1992.

[7] J. Besag (1986), «On the Statistical Analysis of Dirty Pictures», *Royal Statistical Society*, Vol. 48, n° 3, pp 259-302, 1986.

[8] A. De Cesare (1996), *Algorithmes rapides de restauration des signaux : application à l'imagerie médicale*. Thèse de l'Université de Paris-Sud, Février 1996.

[9] B. Chalmond (1988), «Image restoration using an estimated Markov model», *Signal Processing*, Vol. 15, pp. 115-129, 1988.

[10] JP. Cocquerez & S. Philipp (1991), «GDR 134. Traitement du signal et images. Rapport Segmentation. 1ère partie : Prétraitement et approche frontière», *Publication du CNRS, opération GDRS 134.*, (ouvrage collectif de synthèse), 1991.

[11] M. Cosnard & D. Trystram (1993), *Algorithmes et architectures parallèles*, Inter-Editions, 1993.

[12] M. Douimi (1995), *Modélisation markovienne et optimisation numérique pour la restauration des signaux en (1D) et (2D)*. Thèse de l'Université de Rouen, Oct. 1995.

[13] D. Geiger & F. Girosi, (1989) «Parallel and determinist algorithms for MRFs : surface reconstruction and integration», *MIT AI Memo 1114*, Juin 1989.

[14] D. Geman & S. Geman (1984), «Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Image», *IEEE Trans. on Pattern Anal. Mach. Intell.*, Vol. 6, pp. 721-741, nov. 1984.

[15] G. Golub & C.H. Van Loan (1993), *Matrix Computations*, The Johns Hopkins University Press, Baltimore, London, 2 édition, 1993.

[16] R. Horst & H. Tuy (1993), *Global optimization : Deterministic approaches*, Springer-Verlag, Berlin, New York, 2 édition, 1993.

[17] R. Horst & P. M. Pardalos (1995), *Nonconvex optimization and its applications. Handbook of Global Optimization*, Kluwer Academic Publishers, Netherlands, 1995.

[18] M. Nielsen (1994), «Surface Reconstruction : GNCs and MFAs», *Rapport INRIA N°2353*, septembre 1994.

[19] L. S. Jacoby, J. S. Kowalik & J. T. Pizzo (1972), *Iterative Methods for Nonlinear Optimisation Problems*. Prentice-Hall, Inc, New-Jersey, 1972.

[20] S. Kirkpatrick, C. D. Gellatt, & M. P. Vecchi (1983), *Optimization by simulated annealing*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1983.

[21] P. J. Laurent (1972), *Approximation et Optimisation*, Herman, Paris, 1972.

[22] D. Mumford & J. Shah (1985), «Boundary detection by minimizing functionals», *In Proc. IEEE CVPR*, pp. 22, 1985.

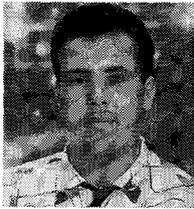
## Algorithme de coupe énumérative

- [23] T. Pavlidis & Y.T. Liow (1990), «Integrating Region Growing and Edge Detection», *IEEE Trans. on PAMI*, Vol. 12, N° 3, pp. 225-233, March 1990.
- [24] W. Snyder, Y. Sik Han, G. Bilbro, R. Whitaker & S. Pizer (1995), "Image Relaxation : Restoration and Feature Extraction". In *IEEE Transaction On Pattern Analysis and Machine Intelligence* , Vol. 17 n° 6, june (1995).
- [25] A. Tikhonov & V. Arsenine (1976), «Méthodes de résolution de problèmes mal posés». Editions MIR , Moscou 1976.

*Manuscrit reçu le 5 décembre 1996.*

### LES AUTEURS

Mohamemd DOUIMI



Mohammed DOUIMI est né en 1965 à Meknès (Maroc). Après des études Universitaires en mathématiques à Fès et à Rabat, il prépare un DEA en Géométrie Algorithmique à l'Université Joseph Fourier de Grenoble. En 1995, il obtient un doctorat en mathématiques appliquées option optimisation numérique préparé au sein du laboratoire de mathématiques et informatique de l'Institut Nationale des Sciences Appliquées (INSA) de Rouen. De 1995 à 97, il occupe un poste d'Attaché Temporaire d'Enseignement et de Recherche (ATER) à l'Université de Rouen. Il est actuellement en vacation à l'Ecole Nationale Supérieure des Arts et Métiers (ENSAM) à Meknès.

Hocine CHERIFI

Hocine CHERIFI est Docteur (NPG) (1984). Maître de conférences habilité. Il est actuellement responsable de l'activité « compression et diagnostic » au sein de l'équipe Ingénierie de la vision du laboratoire TSI de l'Université Jean Monnet. Il s'intéresse au traitement du signal et de l'image, notamment dans le domaine de modélisation mathématique et de la conception d'algorithmes.