

1. Introduction

On considère le problème d'optimisation multi-objective suivant :

$$(P) \begin{cases} \min_y f_1(y), \\ \min_y f_2(y), \end{cases} \quad (1.1)$$

où f_1 et f_2 sont deux fonctions convexes. L'idée est de partager l'ensemble des variables y de minimisation en deux sous ensembles dont les variables seraient U et V . Le but de ce partage serait de minimiser $f_1(U, V)$ par rapport à U en fixant V et de minimiser $f_2(U, V)$ par rapport à V en fixant U , par une méthode d'optimisation concurrente utilisant un algorithme simulant un jeu de Nash entre des joueurs associés aux deux fonctionnelles respectivement. On cherche alors à déterminer $y_{EN} = (U^*, V^*)$ solution du problème suivant

$$(N) \begin{cases} \min_U f_1(U; V^*), \\ \min_V f_2(U^*; V). \end{cases} \quad (1.2)$$

Notre but est de trouver un équilibre de Nash sur le front de Pareto. La difficulté essentielle rencontrée par les algorithmes permettant de calculer un équilibre de Nash [3,9], est la détermination du meilleur partage de la variable y en variables U et V . Dans [4], J.A. Désidéri propose un algorithme de partage de territoire utilisant les vecteurs propres de la matrice Hessienne pour le critère le plus important.

2. Équilibre de Nash

Cette technique d'optimisation provient de la théorie des jeux et consiste à traiter les différents objectifs de manière concurrentielle comme des joueurs indépendants échangeant de l'information partielle. La recherche de l'équilibre de Nash se fait par échange des résultats obtenus pour chaque fonction objectif travaillant avec une partie seulement des variables, les autres étant fixées par les résultats obtenus pour les autres fonctions objectifs. Cet équilibre est atteint quand l'optimisation de chaque fonction conduit toujours à la même solution.

Dans ce qui suit on s'intéresse au cas d'un jeu à deux joueurs. Les deux joueurs cherchent à améliorer leurs propres critères parallèlement. Nous cherchons une stratégie (U^*, V^*) qui définit l'état d'équilibre :

$$\begin{cases} f_1(U^*, V^*) = \min_U f_1(U, V^*), \\ f_2(U^*, V^*) = \min_V f_2(U^*, V). \end{cases} \quad (2.3)$$

La stratégie (U^*, V^*) est un équilibre de Nash.

Nous allons présenter deux algorithmes permettant de définir deux tableaux d'allocations P et Q pour définir les variables du premier joueur notées U et les variables du deuxième joueur V .

3. Algorithmes de partage

Le vrai partage de la variable y est de construire deux tableaux d'allocation P et Q de $\{0, 1\}^n$ avec $P_i + Q_i = 1$ pour $1 \leq i \leq n$ [1]. Soit $I_{12} = \{1, \dots, n\}$ un ensemble d'indices de cardinalité n , I_1 un sous ensemble de I_{12} de cardinalité $n - p$, et I_2 son complémentaire de cardinalité p , c'est-à-dire $I_{12} = I_1 \cup I_2$.

Supposons que :

$$\begin{cases} U = (y_i), & \text{pour } i \in I_1, \\ V = (y_i), & \text{pour } i \in I_2. \end{cases} \quad (3.4)$$

On pose dans ce cas le tableau d'entiers d'allocation P de taille n :

$$P_i = 1, \forall i \in I_1, \quad P_i = 0, \forall i \in I_2,$$

de sorte que

$$y = P.y + (\mathcal{I} - P).y = (U, V) \quad \text{avec } \mathcal{I} = (1, \dots, 1). \quad (3.5)$$

Désignons par "." le produit d'Hadamard, c'est-à-dire $(P.y)_i = P_i y_i$ et $P.y \in \mathbb{R}^n$. Le vecteur (U, V) est défini dans l'équation (3.4).

Le jeu de Nash (N) peut s'écrire sous la forme suivante :

$$(N) \begin{cases} \text{Trouver } y_{EN} \text{ solution de :} \\ \min_U f_1(P.y + (\mathcal{I} - P).y_{EN}), \\ \min_V f_2((\mathcal{I} - P).y + P.y_{EN}), \end{cases} \quad (3.6)$$

On considère le problème (N') suivant :

$$(N') \begin{cases} \text{Trouver } y_{EN} \text{ solution de :} \\ \min_y f_1(P.y + (\mathcal{I} - P).y_{EN}) + f_2((\mathcal{I} - P).y + P.y_{EN}), \end{cases} \quad (3.7)$$

y_{EN} est un équilibre de **Nash** s'il est point fixe de (N') [1].

Pour chaque choix de P on trouve un équilibre de Nash, dans ce cas on a au plus 2^n équilibres de Nash. La question qui se pose est de savoir comment choisir parmi tous ces équilibres, un équilibre appartenant au front de Pareto s'il existe, ce qui n'est pas toujours

le cas en général.

Dans ce travail, nous supposons que les éléments de P appartenant à $[0, 1]$ et nous construisons un autre tableau Q qui va remplacer le tableau $(\mathcal{I} - P)$ de telle sorte que $P + Q$ n'est pas nécessairement égal à \mathcal{I} . Nous avons proposé et expérimenté avec succès deux algorithmes de partage (AG1) et (AG2) qui calculent les deux tableaux P , Q et les équilibres de Nash associé. Dans l'algorithme de partage (AG1) nous avons partagé la variable d'optimisation y , en construisant deux suites de deux tableaux d'allocation $P^{(m)}$ et $Q^{(m)}$ dans $[0, 1]^n$, les deux tableaux sont construits à partir de la minimisation itérative de f_1 et f_2 , et de la minimisation de la fonction f définie par l'équation (3.11). Dans l'algorithme de partage (AG2), les deux tableaux d'allocation P et Q sont considéré comme des variables d'optimisation. Il s'agit de présenter une formulation de jeux de Nash entre deux joueurs en utilisant les deux tableaux d'allocation P et Q , et la variable y .

Dans ce qui suit, nous présentons les deux algorithmes (AG1) et (AG2).

3.1. Algorithme 1 (AG1) : Tableaux d'allocations heuristiques

Soit $m = 0$, à partir d'un point initial $x^{(0)}$ et $y^{(0)}$ quelconque de \mathbb{R}^n , on calcule $P^{(0)}$ et $Q^{(0)}$ par :

$$\left\{ \begin{array}{l} \min_{x \in \mathbb{R}^n} f_1(x), \quad x^{(k+1)} = x^{(k)} - \rho_k \nabla f_1(x^{(k)}), \quad k \geq 0, \\ \\ P_j^{(0)} = \frac{\sum_k |x_j^{(k+1)} - x_j^{(k)}|}{\sum_k \|x^{(k+1)} - x^{(k)}\|}, \\ \\ \min_{y \in \mathbb{R}^n} f_2(y), \quad y^{(k+1)} = y^{(k)} - \rho_k \nabla f_2(y^{(k)}), \quad k \geq 0, \\ \\ Q_j^{(0)} = \frac{\sum_k |y_j^{(k+1)} - y_j^{(k)}|}{\sum_k \|y^{(k+1)} - y^{(k)}\|}, \end{array} \right. \quad (3.8)$$

où ρ_k le paramètre de la méthode du gradient conjugué [2]. Notons,

$$y_{EN}^{(m)} = P^{(m)} \cdot x^* + Q^{(m)} \cdot y^* \quad (3.9)$$

où,

$$\left\{ \begin{array}{l} x^* = \text{Arg} \min_x f_1(x) \\ y^* = \text{Arg} \min_y f_2(y). \end{array} \right. \quad (3.10)$$

On définit la fonction f par

$$f(x) = f_1(\bar{x}) + f_2(\bar{y}) \quad (3.11)$$

où,

$$\bar{x} = P^{(m)}.x + Q^{(m)}.y_{EN}^{(m)} \quad \text{et} \quad \bar{y} = P^{(m)}.y_{EN}^{(m)} + Q^{(m)}.x$$

Etape 2 :

On construit $P^{(m+1)}$ et $Q^{(m+1)}$ tels que

$$\left\{ \begin{array}{l} \min_x f(x), \quad x^{(k+1)} = x^{(k)} - \rho_k \nabla f(x^{(k)}) \quad k \geq 0, \\ P_j^{(m+1)} = \frac{\sum_k |(\nabla f_1(P^{(m)}.x^{(k+1)} + Q^{(m)}.y_{EN}^{(m)}))_j|}{\sum_k \|\nabla f_1(P^{(m)}.x^{(k+1)} + Q^{(m)}.y_{EN}^{(m)})\|}, \\ Q_j^{(m+1)} = \frac{\sum_k |(\nabla f_2(Q^{(m)}.x^{(k+1)} + P^{(m)}.y_{EN}^{(m)}))_j|}{\sum_k \|\nabla f_2(Q^{(m)}.x^{(k+1)} + P^{(m)}.y_{EN}^{(m)})\|}. \end{array} \right. \quad (3.12)$$

$$y_{EN}^{(m+1)} = P^{(m)}.x_{opt}^{(m)} + Q^{(m)}.y_{EN}^{(m)} \quad (3.13)$$

où $x_{opt}^{(m)}$ est la solution du $\min_x f(x)$.

Tant que $\|y_{EN}^{(m+1)} - y_{EN}^{(m)}\| > test$, poser $m = m + 1$, et répéter (3.12).

3.2. Algorithme 2 (AG2) : Tableaux d'allocations optimaux

1. construction de $P^{(0)}$ et $Q^{(0)}$ en utilisant l'étape 1 de (AG1)
2. pour $P^{(m)}$ et $Q^{(m)}$ donnés, minimiser $f(x)$ par rapport à x , on obtient $x_{opt}^{(m)}$
3. minimiser la fonction $f(x)$ par rapport à P , on obtient $P^{(m+1)}$
4. minimiser la fonction $f(x)$ par rapport à Q , on obtient $Q^{(m+1)}$
5. $y_{EN}^{(m+1)} = P^{(m+1)}.x_{opt}^{(m)} + Q^{(m+1)}.y_{EN}^{(m)}$
6. tant que $\|y_{EN}^{(m+1)} - y_{EN}^{(m)}\| > test$.
7. poser, $m = m + 1$, retour à 2.

Notre but à présent est de montrer que le meilleur partage qu'on obtient par les deux algorithmes (AG1) et (AG2), donne un équilibre de Nash qui appartient au front de Pareto ou bien proche du front.

Le front de Pareto est l'ensemble des solutions non dominées[1]. Lorsque le front de Pareto est convexe, on peut trouver tous ses points en minimisant

$$f_w = w f_1 + (1 - w) f_2, \quad w \in [0, 1]. \quad (3.14)$$

Pour tout w , on trouve un point d'équilibre au sens de Pareto.

Dans la partie suivante nous présentons quelques résultats numériques dans le cas où f_1 et f_2 sont convexe, et pour des matrices symétriques définies positives en appliquant (AG1) et (AG2). Une autre application en traitement d'images en appliquant (AG1).

4. Résultats Numériques

On considère deux fonctions convexes de la forme suivante :

$$f_1(x) = \|Ax - b\|^2 \text{ et } f_2(x) = \|Cx - d\|^2, \quad x \in R^{n \times 1} \quad (4.15)$$

où A et C sont deux matrices de taille (n, n) , b et d deux vecteurs de taille $(n, 1)$, et $\|\cdot\|$ est la norme euclidienne : $\|x\| = \left(\sum_{i=1}^n x_i^2\right)^{\frac{1}{2}}$

Dans l'étape 2 de l'algorithme (AG1), pour calculer le gradient de la fonction f nous utilisons les techniques suivantes :

$$\bar{x} = P^{(0)} \cdot x + Q^{(0)} \cdot y_{EN}^{(0)} \quad \text{et} \quad \bar{y} = P^{(0)} \cdot y_{EN}^{(0)} + Q^{(0)} \cdot x. \quad (4.16)$$

A l'itération m de l'étape 2, la fonction f s'écrit sous la forme suivante :

$$f(x) = \|A(P^{(m)} \cdot x + Q^{(m)} \cdot y_{EN}^{(m)}) - b\|^2 + \|C(Q^{(m)} \cdot x + P^{(m)} \cdot y_{EN}^{(m)}) - d\|^2, \quad (4.17)$$

avec,

$$P_m = \begin{pmatrix} P_1^{(m)} & 0 & 0 & \dots & 0 \\ 0 & P_2^{(m)} & 0 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & P_n^{(m)} \end{pmatrix} \quad \text{et} \quad Q_m = \begin{pmatrix} Q_1^{(m)} & 0 & 0 & \dots & 0 \\ 0 & Q_2^{(m)} & 0 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & Q_n^{(m)} \end{pmatrix}.$$

donc

$$\nabla_x f(x) = 2(AP_m)^T (A(P^{(m)} \cdot x + Q^{(m)} \cdot y_{EN}^{(m)}) - b) + 2(CQ_m)^T (C(Q^{(m)} \cdot x + P^{(m)} \cdot y_{EN}^{(m)}) - d),$$

4.1. Test 1

Dans ce test, on utilise les données suivantes :

$$A = C = Id, b = [1; -2; 2; 9; 1; 2; 9], d = [5; 1; 3; -8; -6; 0; 4]. \quad (4.18)$$

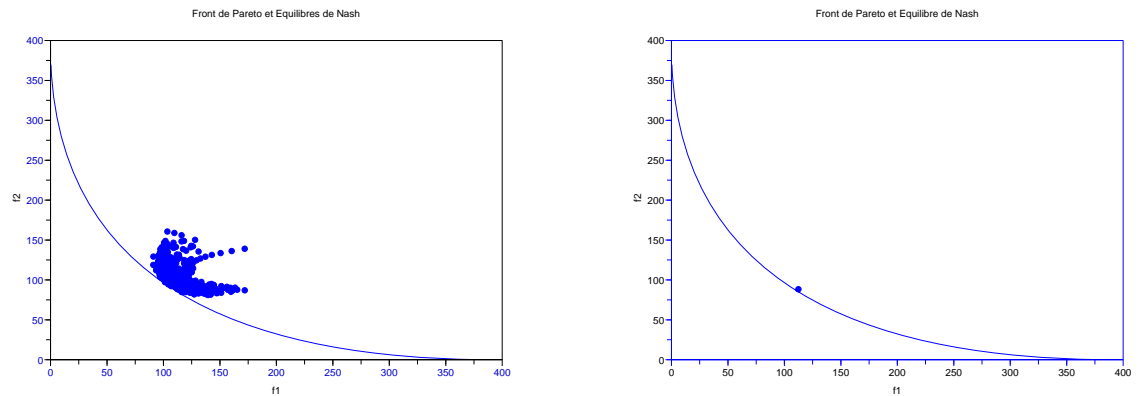


Figure 1 – A gauche, Front de Pareto et équilibres de Nash, A droite, dernier équilibre de Nash trouvé par (AG1)

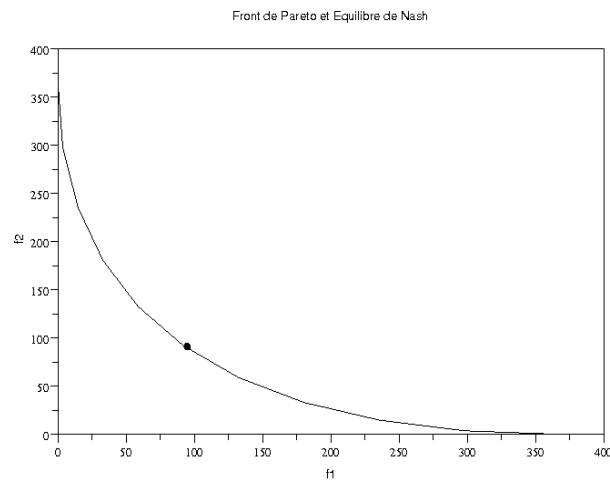


Figure 2 – Dernier équilibre de Nash trouvé par (AG2)

ARIMA

4.2. Test 2

Dans ce test, on utilise les données suivantes :

$$\mathbf{A} = \text{tridiag}[1, -2, 1], \quad C = A; \quad b = \text{rand}(n, 1), \quad d = 10 * b, \quad n = 50 \quad (4.19)$$

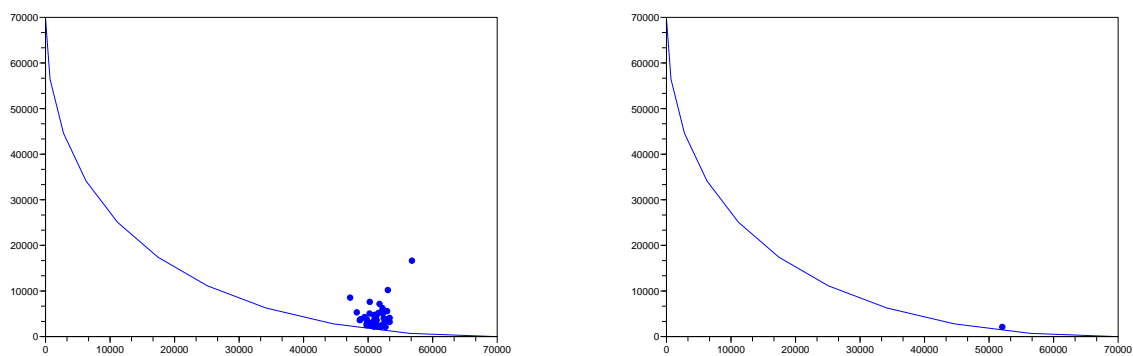


Figure 3 – A gauche, Front de Pareto et équilibres de Nash. A droite, dernier équilibre de Nash trouvé par (AG1).

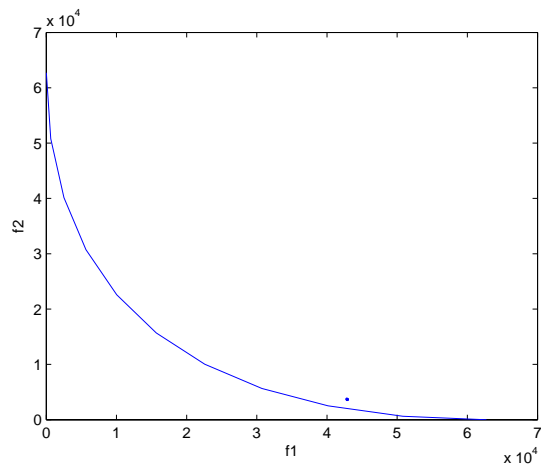


Figure 4 – Dernier équilibre de Nash trouvé par (AG2)

On remarque que le dernier équilibre de Nash trouvé par (AG1) et (AG2) appartient au front de Pareto.

ARIMA

4.3. Application au traitement d'images

Soit Ω un ouvert borné, y une fonction de $H_0^1(\Omega)$ et F une image bruitée supposée dans $L^2(\Omega)$. A partir de l'image observée F on cherche à reconstruire y comme solution du problème de minimisation de :

$$P(y) = \frac{1}{2} \int_{\Omega} |y - F|^2 dx. \quad (4.20)$$

Il s'agit d'un problème inverse mal posé. Pour le résoudre numériquement, on est amené à introduire un terme de régularisation, et à considérer le problème de minimisation de :

$$P_{\varepsilon}(y) = \underbrace{\frac{1}{2} \int_{\Omega} |y - F|^2 dx}_{\text{Ajustement aux données}} + \underbrace{\frac{\varepsilon}{2} \int_{\Omega} |\nabla y|^2 dx}_{\text{Régularisation}} \quad (\text{Tikhonov}) \quad (4.21)$$

où ε est un petit paramètre positif.

La minimisation de la fonctionnelle P_{ε} passe par une diminution du gradient $\int_{\Omega} |\nabla y|^2 dx$ donc un lissage global de la solution[4,10]. Les discontinuités sont de ce fait excessivement pénalisées et l'image estimée a un aspect trop lisse[4,10].

Le coefficient ε permet de jouer sur le poids du terme régularisant par rapport au terme d'ajustement aux données, c'est un paramètre qui, pour une certaine valeur, assure une meilleure restauration de l'image[7,9]. Le choix de ce paramètre est important pour la qualité des résultats. Pour éviter ce choix, nous proposons une autre technique de régularisation utilisant l'optimisation concourante basée sur le partage de territoire entre deux joueurs où le critère d'identification f_1 joue contre le critère de régularisation f_2 tel que :

$$f_1(y) = \frac{1}{2} \|y - F\|^2 + \frac{\lambda}{2} \|\nabla y\|^2 \quad \text{et} \quad f_2(y) = \frac{1}{2} \|\nabla y\|^2 \quad \forall y \in H_0^1(\Omega)$$

où λ est un petit paramètre fixé dans l'intervalle $[0, 1]$, joue ici le rôle d'un paramètre de régularisation.

Parmi les critères quantitatifs les plus courants pour évaluer les performances des modèles de débruitage, nous utilisons le rapport signal-bruit (SNR) [5], elle est exprimée en décibels par la relation entre l'image de référence I_1 et l'image I_2 après analyse :

$$SNR(I_1/I_2) = 10 \log_{10} \left[\frac{\sigma^2(I_1)}{\sigma^2(I_1 - I_2)} \right].$$

Pour obtenir la meilleure solution du problème P_{ε} nous varions le paramètre ε et nous calculons le rapport du signal à bruit (SNR), pour plusieurs valeurs de ε nous avons obtenu les résultats suivants :

Dans les tests qui suivent, on compare les résultats obtenus par la méthode de régularisation de Tikhonov avec ceux obtenus par la régularisation de jeux de Nash en utilisant

l'algorithme 1 (AG1). Pour obtenir la solution optimale par la régularisation de Tikhonov, on sélectionne le bon ε . La figure 5 présente l'évolution de la SNR en fonction ε , avec $\varepsilon = 10^{-i}$, $i = 0 : 9$, nous remarquons que le bon ε est associé à la valeur de la $SNR = 17.7189$, c'est-à-dire $\varepsilon = 0.1$, après les valeurs de la SNR diminuée vers la valeur 15. Par contre dans le cas de Nash les valeurs de la SNR reste proche de la valeur maximale lorsque le paramètre λ varie, voir le tableau 1.

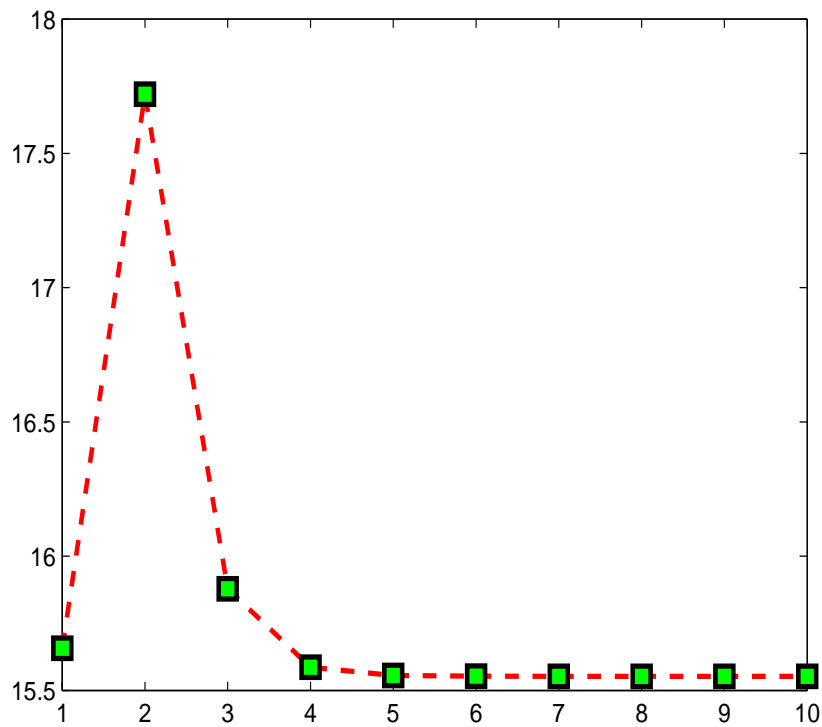
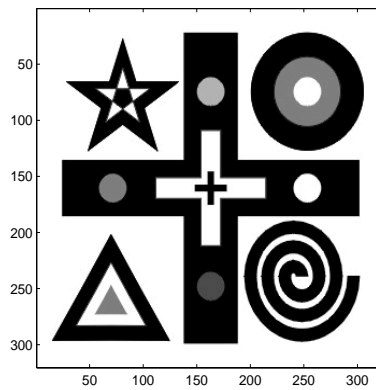


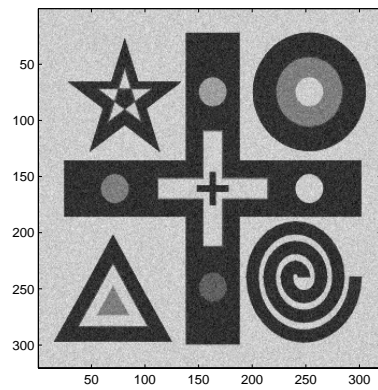
Figure 5 – L'évolution de la SNR en fonction de ε , avec $\varepsilon = 10^{-i}$, $i = 0 : 9$

Tableau 1 – L'importance du partage de la variable d'optimisation

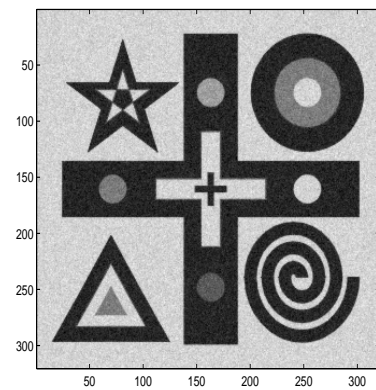
	Tikhonov	AG1			
SNR	17.7189	17.5123	17.7234	16.8802	16.6415
Paramètres	$\varepsilon = 0.1$	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$	$\lambda = 0.0001$



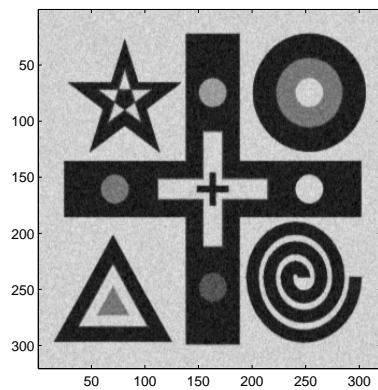
(a) Image originale



(b) Image bruitée



(c) Image restaurée par Tikhonov

(d) Image restaurée par (AG1) ($\lambda = 0.1$)

On remarque que l'algorithme (AG1) donne des résultats robustes par rapport aux variations de λ contrairement à Tikhonov, donc notre approche ne dépend pas au choix de λ .

5. Conclusion

Dans ce travail nous avons présenté deux algorithmes de partage de variable d'optimisation, le premier construit deux tableaux d'allocations heuristiques et l'autre construit deux tableaux d'allocations optimaux. Après la convergence, on obtient un équilibre de Nash appartenant au front de Pareto dans le cas des exemples considérés. Les algorithmes proposés permettent d'affecter à chaque fonction $f_i, i = 1 : 2$ sa variable d'état à l'aide des tableaux d'allocation P et Q construits automatiquement. Les premiers résultats obtenus par l'algorithme (AG1), pour le traitement d'image sont encourageants.

Remerciements

Ce travail a été supporté par l'Action Intégrée Franco-Morocaine MA/05,115 et le Programme SARIMA.

6. Bibliographie

- [1] Aboulaich R., Habbal A., Moussaid N., *Split of an optimization variable in game theory*, *Math. Model. Nat. Phenom(MMNP)*, accepted
- [2] Allaire G. *Conception optimale de structures*, *Mathématiques et Applications*, Springer Verlag, Heidelberg, 2006
- [3] Aubin J.P., *Mathematical methods of game and economic theory*. North-Holland Publishing Co. Amsterdam, New York, 1979.
- [4] Désidéri J.A., *Split of Territories in Concurrent Optimization*. Rapport de recherche, Institut National de Recherche en Informatique et en Automatique , N° 6108, 8th October 2007.
- [5] Destuynder Philippe, *Cours d'Analyse, Traitement et Synthèse d'Images Numériques*. Calcul Scientifique CNAM Paris 292 rue saint Martin 75003.20 mars 2005
- [6] El Guarmah E., Sellami H., *A comparison between a linear approach and a nonlinear diffusion model in image restoration*. IJAM vol. 22, n° 1, 2009
- [7] A. Habbal, J. Petersson and M. Thellner, *Multidisciplinary topology optimization solved as a Nash game*. Int. J. Numer. Meth. Engng 2004 ; 61 :949-963
- [8] P. Destuynder, M. Jaoua, and H. Sellami, *A dual algorithm for denoising and preserving edges in image processing*. J. Inv. Ill-Posed Problems 15 (2007), 19 – 35.
- [9] A. Habbal, H. Lievre, J.A Desideri, *Optimisation numérique de profils d'aile par algorithmes génétiques et jeux de Nash*. Reserach report number 4275, INRIA, septembr 2001.
- [10] Jean-François Aujol, *Contribution à l'analyse textures en traitement d'images par méthodes variationnelles et équations aux dérivées partielles..* Thèse, Université de Nice-Sophia Antipolis, 2004.

- [11] Tikhonov, A.N., Goncharsky, A.V., Stepanov, V.V., Yagola, A.G., *Numerical methods for the solution of ill-posed problems*,. Kluwer Academic Publishers, 1995.