

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed Boudiaf University of M'sila
Faculty of Technology



الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة المسيلة
كلية التكنولوجيا

Département d'Électronique

MEMOIRE DE FIN D'ETUDE

En vue de l'obtention du diplôme de :

MASTER

En Électronique

Option : Système de télécommunication

Présenté par :

Saghiour Soulaf & LAmrai Ahlam

Thème

**SEPARATION AVEUGLE DE SOURCES (SAS) PAR
OPTIMISATION HEP SO (HIGH EXPLORATION PARTICULE
SWARM OPTIMISATION) : APPLICATION AUX SIGNAUX
IMAGES**

Devant le jury composé de :

NOM et Prénom	Grade	Qualité
LALAOUI LAHOUAOUI	MCA	Président
KHALFA ALI	MCB	Encadreur
SAHED MOHAMED	MCA	Examineur

Année Universitaire : 2021 / 2022

N° d'ordre : GM/...../2022

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciement

Je tiens tout d'abord à remercier vivement mon Rapporteur, **Mr. KHALFA Ali**, docteur à l'université Mohamed Boudiaf de M'sila, pour avoir dirigé et encadré cette mémoire, ainsi que pour le temps passé à me conseiller. Je le remercie aussi pour sa confiance, sa souplesse et son ouverture d'esprit qui m'ont permis de mener à bien ce travail de recherche.

Je tiens à exprimer ma profonde gratitude à **Mr. Lahouaoui Lalaoui** docteur à l'université Mohamed Boudiaf de M'sila, pour m'avoir fait l'honneur de présider le jury de notre master et pour l'intérêt et le soutien dont il m'a fait preuve.

Mes remerciements vont aussi à **Mr. SAHED MOHAMED** i docteur à l'université Mohamed Boudiaf de M'sila, pour avoir accepté d'évaluer et de juger mon travail en tant que membres de jury de notre mémoire master.



Dédicaces

Je dédie ce modeste travail :

A ma très chère mère et A mon très cher père
Pour leur amour, sacrifices et encouragements.

A tous les gens qui m'aiment

A tous ceux qui m'ont aidé

Ahlam

*They say a person needs just three things to be truly happy in this world:
someone to love, something to do, and something to hope for.*

Dédicaces

Entier sur une échelle et qu'on le mettait de l'autre côté, sa paume se
balancerait et l'autre côté s'envolerait

Et à mon cher père, le sommet de ma tête, que Dieu me le garde, et que
Dieu les perpétue comme une lumière qui illumine nos vies de leur
présence

Et à mes frères qui m'ont soutenu tout au long de mon parcours
académique, et à ma chère soeur

Et à tous les honorables membres de ma famille, mes collègues

Et tous ceux qui m'ont soutenu tout au long de mon parcours académique

Soulaf

*They say a person needs just three things to be truly happy in this world:
someone to love, something to do, and something to hope for.*

Résumé

La Séparation Aveugle de Sources (SAS), qui consiste à estimer un ensemble de signaux sources inconnus à partir d'un ensemble de signaux observés qui sont des mélanges à paramètres inconnus de ces signaux sources {de sorte que les composants des sources observées sont indépendamment indépendants les uns des autres ou sans rapport les uns avec les autres}. Afin de les évaluer et de les développer en comparant leurs performances sur la base d'un ensemble d'expériences de séparation aveugle des sources de signaux utilisées (images).

Dans la deuxième partie, l'algorithme a été développé HEPSON où cela dépend de deux facteurs comme suit : le premier opérateur dépend d'un mécanisme d'intersection multiple pour l'algorithme génétique ; tandis que le second opérateur dépend du mécanisme de la colonie d'abeilles, les deux opérateurs sont utilisés pour mettre à jour l'emplacement et la vitesse des molécules afin d'obtenir la matrice de séparation optimale FastICA. Et moderne basé sur HEPSON

ملخص

يُعد الفصل المكفوف للمصدر (BSS), والذي ينطوي على تقدير مجموعة غير معروفة من اشارات المصادر انطلاقا من الإشارات الملحوظة والتي هي مزيج من اشارات المصادر {بحيث تكون فيها مكونات المصادر التي تمت ملاحظتها مستقلة بشكل متبادل او غير مرتبطة ببعضها البعض}. الهدف من هذه المذكرة هو تقديم خوارزميات BSS من اجل تقييمها وتطويرها وذلك بمقارنة ادائها استنادا الى مجموعة من تجارب الفصل المكفوف لمصادر الاشارات المستخدمة (الصور).

في الجزء الثاني تم تطوير الخوارزمية HEPSON حيث تعتمد على عاملين كالتالي : المشغل الاول يعتمد على الية تقاطع متعددة للخوارزمية الجينية؛ في حين المشغل الثاني يعتمد على الية مستعمرة النحل , يتم استخدام كلا المشغلين بهدف تحديث موقع وسرعة الجزيئات من اجل الحصول على مصفوفة الفصل الامثل . في الاخير وبالمقارنة مع الطرق الموجودة تم الحصول على نتائج ممتازة باستعمال طريقتان كلاسيكية متمثلة في FactICA والحديثة استناد ل HEPSON

Abstract

Table des Matiere

Remerciement	_____
Dédicaces	_____
Résumé	_____
Introduction Générale	_____ 1
CHAPITRE I ETAT DE L'ARTSUR LA SEPARATION AVEUGLE DE SOURCES	
I.1 Introduction	_____ 5
I.2 Historique	_____ 5
I-3 Différents types de mélanges	_____ 5

I.3.1. Les mélanges linéaires instantanés :	5
I.3.2. Les mélange convolutifs :	6
I.3.3. Mélange non linéaires :	7
I.3.3.1 Mélange post non linéaire (PNL) :	7
I-4. Méthodes utilisés en séparation aveugle source(SAS)	8
I-4.1 Analyse en composantes principales (PCA) :	8
1.4.2 Analyse en Composantes Indépendantes (ICA) :	8
1.4.3 Minimisation de l'information mutuelle	8
1.4.4 Principe de base de l'algorithme Fast-ICA	9
I.5 applications de la séparation aveugle de sources :	9
I.5.1 Séparation du signal vocale	9
I.5.2 Traitement et reconnaissance d'images	9
I.5.3 Traitement du signal biomédical	10
I.5.4 Télécommunication	10
I.6 Conclusion	11

CHAPITRE II Traitement d'image

II.1 Introduction :	13
II.2 Historique :	13
II.3 Définition d'une image :	14
II.3 .1 Image analogique	14
II.3 .2 Image numérique :(numérisée) :	14
II.3 .3 Acquisition des Images Numériques :	14
II.4 Caractéristiques d'une image numérique	15
II.4. 1 Dimension	15
II.4. 2 Résolution	15
II.4. 3 Bruit	16
II.4. 4 Histogramme	16
II.4. 5 Images à niveaux de gris	17
II. 5 Types d'images	18
II. 5.1 Images binaires (en noir et blanc)	18
II. 5.2 Images à niveaux de gris (Monochromes)	19
II. 5.3 Images en couleurs (Polychromes)	20
II. 6 qualités de l'image numérique	21
II. 7 Système de traitement d'image	21

II. 8 Les différents formats d'images	22
II. 8 .1. Images matricielles (Bitmap)	22
II. 8 2. Images vectorielles	23
II. 8 .3. JPEG	24
II. 8 4. TIFF	24
II. 8 .5 GIF	24
II. 8 .6. PNG	25
II. 9 Domaines d'application	25
II. 10 Conclusion	25
III.1 Introduction	27
III .2 - Définition de l'optimisation :	28
III .3 Les algorithmes évolutionnaires :	28
III .3.1 Principe des algorithmes évolutionnaires :	29
III .4 Définitions et notations :	29
III .4.1 Le squelette :	30
III .5 La recherche tabou :	31
III.5.1 Concepts de base de la recherche taboue :	31
III.6.Algorithme d'optimisation basée sur la biogéographie :	32
III .7 L'algorithme d'ascension de collines (Hill Climbing) :	37
III.8 L'algorithme du Recuit Simulé :	40
III.8.1 Interprétation	42
III.9 Algorithmes génétiques :	42
III.9.1 Terminologie et éléments de base :	43
III.9.2 Évolution des espèces	44
III.9.3 Conception d'un algorithme génétique	45
III.9.4 Comment fonctionne l'algorithme génétique	46
III.9.5 Variantes	47
III.9.5.1 Codage	47
III.9.5.1.a Codage binaire	47
III.9.5.1.b Codage réel	48
III.9.5.2 Évaluation : condition physique (fitness)	48
III.9.5.3 Population initiale	48
III.9.5.4 Critère d'arrêt	49
III.9.5.5Sélection	49
III.9.5.6Croisement	50
III.9.5.6.1 Croisement binaire	51
III.9.6 Transfert	52

Fig. II. 5 Image au niveau de gris. [7].....	18
Fig. II. 6 Image binaire	19
Fig. II. 7 Image Monochrome (256 couleurs).....	20
Fig. II. 8 Image polychrome (65536 couleurs).....	21
Fig. II. 9 Composition d'un système de traitement numérique.	22
Fig. II. 10 Images matricielles [10].....	23
Fig. II. 11 Images vectorielle [10].....	24

Chapitre III

Fig. III. 1 Classification générale des méthodes d'optimisation.....	28
Fig. III. 2 Squelette d'un algorithme évolutionnaire [III.10].....	30
Fig. III. 3 Modèle linéaire de taux d'émigration (μ) et le taux d'immigration (λ) [III.16]	34
Fig. III. 4 Schéma de principe de base de l'algorithme Hill Climbing (HC)	39
Fig. III. 5 Cycle génétique.....	46
Fig. III. 6 Les phases de définition d'un codage	47
Fig. III. 7 Croisement en un point de deux chromosomes.....	51
Fig. III. 8 Croisement en 2-points de deux chromosomes.	51
Fig. III. 9 Croisement uniforme	52

Chapitre IV

Fig. IV. 1 Schéma de la méthode proposée.....	57
Fig. IV. 2 Simulations_1 : Lena _ Barbar	Erreur ! Signet non défini.
Fig. IV. 3 Simulations_2 : ily _ Einstein	Erreur ! Signet non défini.

Liste des Tableaux

Chapitre III

Tableaux. III. 1 Algorithme de la recherche tabou (TS).....	32
Tableaux. III. 2 Tableau III.2 : L'algorithme BBO.....	36
Tableaux. III. 3 . L'Algorithme du Hill Climbing (HC).....	38
Tableaux. III. 4 L'algorithme d'optimisation Recuit Simulé	41
Tableaux. III.5 . Comparaison de la terminologie naturelle et celle des algorithmes génétiques	43

Chapitre IV

Tableaux. IV. 1 Les signaux source de l'image test [IV.8]	59
Tableaux. IV. 2 PSNR (dB) pour l'algorithme Fast_ICA et HEPSO	67
Tableaux. IV. 3 ssim (dB) pour l'algorithme Fast_ICA et HEPSO	67
Tableaux. IV. 4 AVCC pour le HEPSO et l'algorithme Fast_ICA	69
Tableaux. IV. 5 AVCC pour le HEPSO et l'algorithme Fast_ICA	69

Introduction Générale

Il s'agit dans le cadre de cette mémoire de développer des méthodes de Séparation Aveugle de Sources (SAS) par optimisation HEPSO applicables à des images.

La Séparation Aveugle de Sources (SAS) consiste, à partir de signaux donnés qui sont des mélanges, à estimer les signaux sources qui sont à l'origine de ces mélanges. Cette estimation se fait sans beaucoup d'informations a priori sur les sources et les paramètres de mélange, d'où le terme "aveugle".

Le terme aveugle signifie que la séparation est faite sans information a priori sur les signaux sources, la seule information exigée est l'indépendance statistique de ces signaux.

La SAS peut être appliqué à une variété de situations telles que l'analyse des signaux biomédicaux obtenus par EEG ou dans les télécommunications sans fil pour séparer plusieurs signaux reçus. Elle est aussi appliquée dans la séparation de sons en diaphonie et dans la suppression du bruit additif des signaux et des images. Actuellement, au niveau mondial, les applications se concentrent sur trois domaines : le biomédical, l'audio et les télécommunications.

Dans ce travail, on présente une étude rigoureuse du problème de la séparation aveugle de sources. Plusieurs techniques ou approches sont présentées également par des chercheurs pour apporter une solution à ce problème à savoir des techniques basées sur les réseaux de neurones artificiels, sur la maximisation et minimisation de l'information, sur l'information mutuelle, et sur les statistiques d'ordres supérieurs. On s'est focalisé sur le choix des approches neuronales dites également méthodes connexionnistes. Ce choix est légitime car le premier algorithme proposé était basé essentiellement sur une méthode connexionniste d'un côté, d'un autre côté, plusieurs chercheurs se sont concentrés sur cet axe.

Dans ce manuscrit, on expose quelques algorithmes adaptatifs de séparation aveugle de sources. Afin de démontrer les performances et les qualités des différents algorithmes neuronaux utilisés, un programme en MATLAB étant implémenté pour faciliter cette démarche.

Le but de cette thèse est d'exploiter et aussi de développer des algorithmes qui simulent les éléments de la matrice de séparation pour avoir une matrice optimale qui rend les sources indépendantes.

Ce mémoire est articulé sur quatre chapitres :

Dans le premier chapitre on rappelle l'état de l'art de la SAS. Nous commençons par donner une description générale de la SAS, les types de mélanges et leurs modèles mathématiques, ainsi que les applications de la SAS.

Dans le deuxième chapitre, nous introduisons brièvement le domaine du traitement d'images. Une synthèse bibliographique sur les différentes approches de segmentation d'images est présentée.

Le chapitre trois traite plusieurs approches de séparation basées principalement et essentiellement sur les techniques neuronales, introduit les différents algorithmes évolutionnaires existant dans la littérature et abordés dans cet axe de recherche. Nous les présentons en détails en insistant sur leur rôle d'éviter de tomber dans les minima locaux.

Le quatrième chapitre décrit une nouvelle méthode d'optimisation évolutionnaire, notée HEPSO (High Exploration Particle Swarm Optimization) et expose les résultats de quelques exemples de simulation pour chaque algorithme étudié afin de donner les avantages et les limitations de chaque approche.

Une conclusion générale donnera une synthèse en ce qui concerne les différentes démarches suivies dans ce travail et expose également les perspectives envisagées pour les travaux futurs.

CHAPITRE I
ETAT DE L'ART
SUR LA SEPARATION
AVEUGLE DE SOURCES

I.1 Introduction

La séparation à la source aveugle (SAS) est une méthode relativement nouvelle sujet en traitement du signal qui a été initié au milieu des années 80 par les travaux de 'Herault et Jutten ', Le problème consiste à récupérer sources originales à partir de mélanges d'entre eux, en supposant qu'il n'y a d'informations ni sur les signaux source d'origine, ni sur le système de mélange (d'où le terme aveugle).

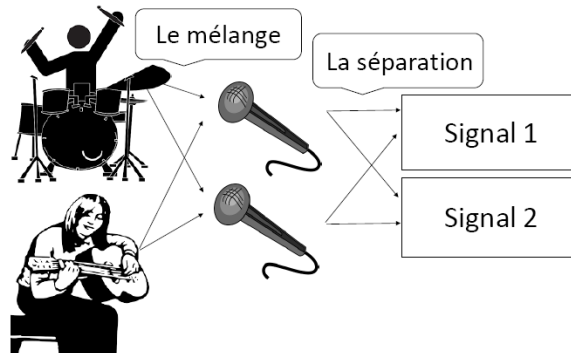


Fig. I. 1 Illustration d'un processus de séparation de deux sources à partir de deux observations.

I.2 Historique

Le premier travail dans le domaine de séparation aveugle de source a été présenté par Héroult et Jutten dans les années 80 [I.1]. Taleb et Jutten décrivent le problème biologique qui consiste à des réponses musculaires émises à l'essai de différentes sorties d'excitation [I.2]. Les travaux de P. Comon en 1994 ont permis de formaliser le lien entre la méthode d'analyse en composantes indépendantes (ICA) et le problème de la séparation aveugle de source (BSS) dans le cadre d'un mélange linéaire. [I.3][I.4]. En 1999, Taleb et Jutten ont introduit un nouveau type de mélange, les mélanges post non linéaire. [I.5]. Les travaux de J.F. Cardoso proposent d'envisager une technique appelée the three easy routes to independent component analysis » pour les mélanges linéaires [I.3].

I-3 Différents types de mélanges

Il existe plusieurs types de mélanges divisés en deux classes : les mélanges linéaires et les mélanges non-linéaires

I.3.1. Les mélanges linéaires instantanés :

Le modèle de mélange linéaire instantané. Sous forme matricielle le modèle liant P

observations à N sources est décrit de la manière suivante :

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{1.1}$$

où $\mathbf{x}(t)$ désigne le vecteur d'observations $[x_1(t) \dots x_p(t)]^T$, \mathbf{A} ,la matrice des coefficients constants du mélange et $\mathbf{s}(t)$ le vecteur sources $[s_1(t) \dots s_N(t)]^T$.

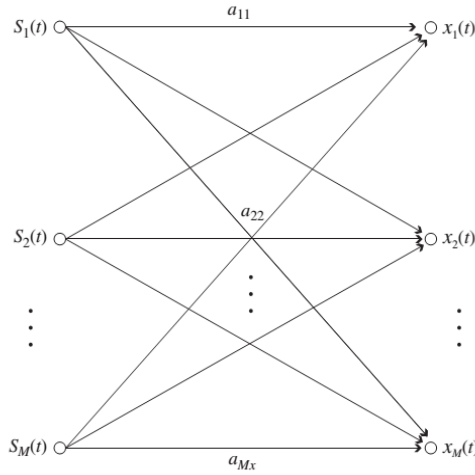


Fig. I. 2 Modèle du mélange linéaire

I.3.2. Les mélange convolutifs :

Le modèle de mélange par convolution est plus proche d'un modèle de mélange réel. On suppose qu'il existe N sources statistiquement indépendantes $s_i(t)$, $i = 1, 2, \dots, N$ reçues par P capteurs après le processus de mélange par convolution [I.6]. Les signaux d'observations sont notés par $x_j(t)$, $j = 1, 2, \dots, P$. Et le modèle de mélange de convolution peut être exprimé comme suit :

$$x_j(t) = \sum_{i=1}^P a_{ji}(t) * s_i(t) \tag{1.2}$$

où * désigne l'opération de convolution et $a_{ji}(\tau)$ est la réponse impulsionnelle de la $j^{ème}$ capteur sur la $i^{ème}$ source. Étant donné que chaque canal peut être désigné par une réponse impulsionnelle finie (FIR) d'ordre L filtre.

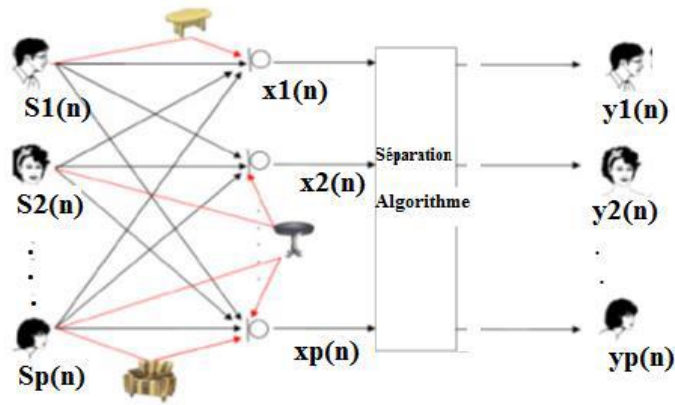


Fig. I. 3 Modèle des mélanges convolutifs

I .3. 3. Mélange non linéaires :

En réalité, les signaux observés sont pour la plupart mélangés de manière non linéaire. Le modèle de mélange non linéaire est une extension du modèle linéaire.

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) + \mathbf{n}(t) \tag{1.3}$$

où $\mathbf{x}(t)$ sont des vecteurs de signaux observés à P dimensions, $\mathbf{s}(t)$ sont à N dimensions des vecteurs de signal source inconnus, et $\mathbf{n}(t)$ sont des vecteurs de bruit additif à P dimensions qui sont indépendants des signaux sources. $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^P$ est une fonction de mélange non linéaire. Ce type de mélange se rapportent à des applications pour lesquelles le modèle linéaire est insuffisant pour décrire le mélange des signaux d'intérêts. On distinguera les deux modèles suivants :

I .3. 3.1 Mélange post non linéaire (PNL) :

Pour ce type de mélange peut se décomposer en une partie linéaire où les sources sont mélangées de manière linéaire instantanée, suivie d'une non-linéarité sur chaque observation. Certains auteurs ont proposé des méthodes pour résoudre le problème ils ont utilisé maximum de vraisemblance [I.7] ou l'information mutuelle [1.8]. Le modèle PNL peut s'exprimer pour N sources et N capteurs comme suit:

$$x_i = f_i(\sum_{j=1}^N a_{ji} \cdot s_j), \quad i = 1 \dots N \tag{1.4}$$

Où x_i les représentent les observations, les f_i sont des fonctions non-linéaires inversibles inconnues, les a_{ji} représentent les coefficients de mélange réels de la partie linéaire instantanée du modèle et les s_j les sources supposées statistiquement indépendantes.

I-4. Méthodes utilisés en séparation aveugle source(SAS)

I-4.1 Analyse en composantes principales (PCA) :

Le principe de PCA est de trouver des variables décorréées. L'intérêt de l'analyse en composantes principales réside dans la réduction de dimensions de représentation de donnée ou la projection des données sur un plan représentatif, ce plan est de maximum d'information [I.15].

1.4.2 Analyse en Composantes Indépendantes (ICA) :

ICA (Independent Component Analysis) est une méthode SAS (séparation aveugle source) qui trouve des composants statistiquement indépendants et non gaussiens à partir des données. Les premières méthodes de SAS ont été des méthodes basées sur l'ICA [I.16] [I.17] [I.18].

Cette méthode peut être vue comme un prolongement de l'analyse en composantes principales (PCA) [I.19], dans laquelle le critère de non corrélation des sorties est remplacé par l'indépendance statistique. Le développement de la théorie de l'ICA remonte au début des années 1990, lorsque Les chercheurs français Jutten et Héroult (1991) ont d'abord proposé le concept ICA.. L'objectif de l'ICA est d'estimer les sources de telle sorte que celles-ci soient mutuellement statistiquement indépendantes. Cependant, cette hypothèse sur les propriétés des sources n'est pas suffisante pour garantir la séparation des sources.

En effet, il a été démontré dans [I.20] que séparer des signaux indépendants et identiquement distribués (i. i. d) suivant une loi gaussienne est impossible. La résolution du problème de SAS passe par l'ajout d'une des deux hypothèses complémentaires suivantes : soit les sources suivent une loi de probabilité non gaussienne (sauf au plus une), soit les sources (i.i.d). On aboutit donc à deux catégories de méthodes ICA basées sur l'une ou l'autre de ces Hypothèses complémentaires.

1.4.3 Minimisation de l'information mutuelle

Une autre approche pour l'estimation de l'ICA, inspiré par la théorie de l'information (*MI*), est la minimisation de l'information mutuelle. Utilisant le concept de la dérivée de l'entropie, on définit l'information mutuelle I entre N variables aléatoires si avec $i: =...N$ comme suit :

$$I(y_1, y_2 \dots y_M) = \sum_i^N H(y_i) - H(y) \quad (1.5)$$

MI est une mesure de l'indépendance entre les variables aléatoires. *MI* est toujours non négatif, et nul si et seulement si les variables sont statistiquement indépendantes.

1.4.4 Principe de base de l'algorithme Fast-ICA

L'algorithme Fast-ICA a été développé en 1997 par Oja et Hyvarinen [I.24].

Il est aussi l'algorithme de l'ICA le plus couramment utilisé pour la séparation de mélanges instantanés de sources. "Fast-ICA" met en évidence des liens forts entre l'analyse en composantes indépendantes et la poursuite de projection. Cette dernière est une méthode d'analyse des données qui relève des statistiques.

Elle a été initialement proposée par Jérôme H. Friedman and John Tukey [I.25] en 1974. Fast-ICA est un algorithme très performant de maximisation de la fonction de contraste pour les sources non gaussiennes [I.26], . Il est basé sur le principe de l'algorithme d'apprentissage itératif de type point fixe (Fixed point algorithm).

I.5 applications de la séparation aveugle de sources :

I.5.1 Séparation du signal vocale

L'idée de base est qu'il existe plusieurs signaux observés, qui sont mélangés par de nombreux signaux sources, et le but du BSS L'algorithme consiste à séparer les signaux source. Dans un système de reconnaissance vocale.

Le signal observé est la convolution des réponses impulsionnelles produites par l'ensemble du signal vocal source, du capteur et de l'environnement. Dans des situations réelles, les positions des microphones par rapport aux sources arbitraires, le processus de mélange n'est pas connu, et a donc à estimer aveuglement. Dans cette situation, les algorithmes BSS sont importants.

De plus, les algorithmes (SAS) convolutifs ont tendance à être appliqués de préférence dans domaine musique. Douglas (2002) a séparé deux voix en utilisant le BSS convolutif techniques, tandis que Vincent (2005) a appliqué la technologie ICA pour obtenir les sons de différents instruments dans une bande sonore.

I.5.2 Traitement et reconnaissance d'images

Dans le domaine du traitement d'images l'extraction de caractéristiques d'images, la reconnaissance de visages, la détection d'objets en mouvement, le tatouage d'images

numériques, le débruitage d'images, etc. [I.57] [I.58].

Par exemple, les méthodes SAS peuvent aider à récupérer le contenu d'origine d'images de celles contaminées et éliminer les mauvais effets [I.59].

I.5.3 Traitement du signal biomédical

La SAS peut être utilisé pour séparer des signaux d'Electrocardiogramme (ECG), d'Electromyogramme (EMG) ou encore d'Electroencéphalogramme (EEG) [1.44].

Il est possible d'extraire grâce à la SAS les battements cardiaques du fœtus, à partir de plusieurs signaux ECG (ECG fœtus, ECG mère et le bruit qui peut être, par exemple, la respiration ou toute activité musculaire, etc.) reçus au niveau des capteurs placés sur le ventre de la mère [1.45].

Dans [1.46], les auteurs ont proposé une nouvelle méthode basée sur Décomposition en Valeurs Singulières (SVD) et de SAS exploitant la propriété de cyclo-stationarité des signaux, ce qui a permis d'améliorer la précision d'extraction des signaux ECG par rapport à la technique fondée sur l'ACI proposée par [1.45].

I.5.4 Télécommunication

La SAS a une large application dans la communication numérique sans fil (Zhang et Hu, 2000 ; Cances, Mohammadkhani et Meghdadi, 2006).

Divers savants ont proposé utilisant un système de réception de réseau d'antennes et des techniques de filtrage spatial ainsi que un algorithme BSS pour réaliser la séparation du signal multi-utilisateur co-canal, de sorte qu'un plusieurs signaux d'utilisateur peuvent être transmis sur le même canal en même temps, améliorant ainsi considérablement la capacité du canal.

L'application aux systèmes de radiocommunication et plus particulièrement la téléphonie mobile SDMA (Spatial Division Multiple Access). Xavier et al. Adoptent dans [1.42] une modélisation de type SAS pour les GSM (Global System for Mobile) (système MIMO).

Il en va de même en téléphonie à main libre, le signal de parole du locuteur peut être bruité par différentes sources de bruit (musique, voitures, etc.). Dans ce cas-là, la SAS est utilisée à des fins de débruitage de la source utile (en l'occurrence la "parole" du locuteur [1.43]).

I.6 Conclusion

Dans ce chapitre, nous avons donné un historique et des généralités sur le problème de la séparation aveugle de sources (SAS) ainsi que des modèles et des méthodes de mélange linéaires (instantanés et convolutifs) et non linéaires instantanés. Nous avons proposé quelques applications de la (SAS).

CHAPITRE II

Traitement d'image

II.1 Introduction :

L'image, dans un premier temps, se présentait sur un support photographique qui permettait le traitement en différé du phénomène bref, une analyse fine des phénomènes enregistrés et bien sûr l'archivage et l'illustration.

Le traitement d'image est né de l'idée et de la nécessité de remplacer l'observateur humain par la machine. L'image ou les signaux provenant de capteurs ont alors été numérisés pour pouvoir être traités par l'ordinateur. Dans un deuxième temps, l'image a été codée et mémorisée sur différents supports.

Une image est une représentation bidimensionnelle d'une scène. Elle dépend de la scène (géométrie), du système optique (lentilles.), des propriétés associées à la lumière.

Le traitement d'images numériques l'ensemble des techniques permettant de transformer une image numérique afin d'améliorer qualité ou d'en extraire des informations.

II.2 Historique :

Le traitement d'images est commencé dans les années 1920: la presse doit envoyer des informations image entre Londres et New-York ,→ système Bartlane pour le codage d'images (5 niveaux) [II.1] .Le véritable démarrage du traitement d'images que dans les années 1960 quand les ordinateurs commencent à être suffisamment puissants pour travailler pour diffusion d'images.

Fin des années 60, début des années 70 : imagerie médicale (tomographie)

Depuis les années 70 : augmentation constante de l'utilisation des images ,→ géographie, biologie, médecine, nucléaire, internet, télévision, satellite, microscopie,

Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson Prentice Hall, 2008 .
Dans les années 1980, la compréhension d'images par des systèmes experts [II.2]

Dans les années 1990, les opérateurs s'amélioraient constamment. La recherche médicale devient un très gros demandeur en traitement d'images pour améliorer les diagnostics faits à partir des nombreuses techniques d'imagerie médicale, la technique reine étant l'IRM.

Les publicitaires, puis le grand public se familiarisent avec la retouche d'image grâce au logiciel Photoshop, et le traitement d'images dans un objectif esthétique se répand avec l'apparition d'autres logiciels dédiés (The Gimp, Paint Shop Pro).

Enfin, la décennie s'achève sur ondelettes et les images multimodales [II .3]

II.3 Définition d'une image :

Une image est représentée par fonction continue : $f(x,y), x,y \in \mathfrak{R}$ ou par une fonction numérique $f(i,j), i,j \in \mathcal{N}$ après numérisation.

II.3 .1 Image analogique

Elle peut être défini sous la forme d'une fonction $I(x, y)$ analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation. [II .4]

M Sandeli, traitement d'images par des approches bio-inspirées application à la d'images, université constantine 2. 2014.

II.3 .2 Image numérique :(numérisée) :

La conversion I Image analogique vers une image numérique numérisation ce fait en en deux étapes 1 échantillonnage spatial : discrétisation des coordonnées de l'image réelle ;

Etape 2 quantification des luminances : discrétisation des intensités de l'image réelle.

Une image numérique est composée d'un ensemble fini d'éléments, appelés picture element, ou pixels (voxels en 3D) L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs. [II .4]

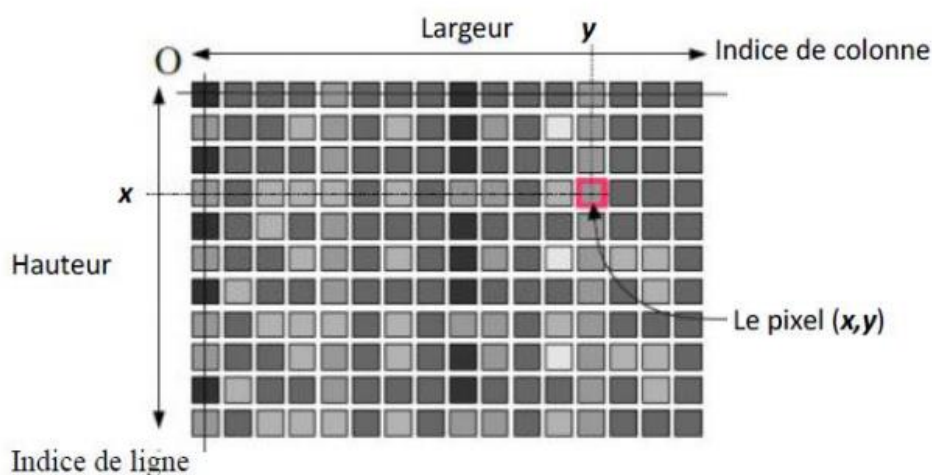


Fig. II. 1 Représentation d'image numérique.

II.3 .3 Acquisition des Images Numériques :

L'acquisition des images est réalisée par un capteur sensible à la lumière et capable de traduire ce signal lumineux en un signal électrique quantifiable.

Plusieurs moyens d'acquisition caméra, microscope, tomographie, infra-rouge, satellite, ... Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device).

La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier [II .5]

Le processus général pour l'acquisition d'une image numérique est montré sur la figure suivante :

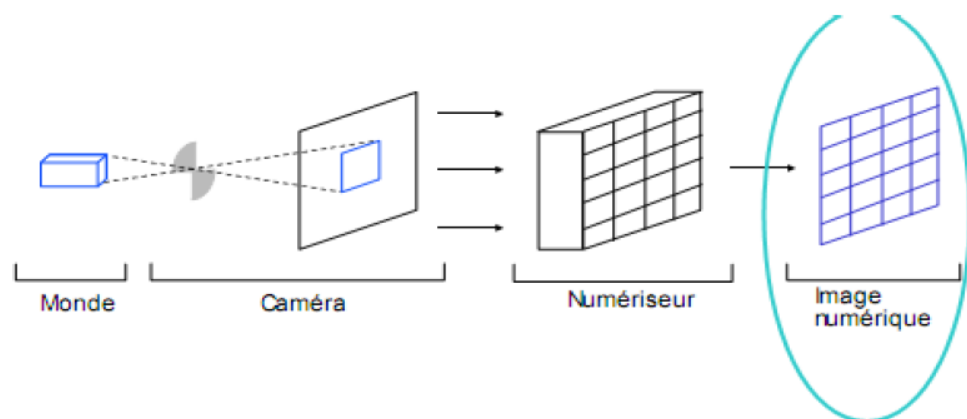


Fig. II. 2 l'acquisition d'une image numérique

II.4 Caractéristiques d'une image numérique

II.4. 1 Dimension

La taille d'une image numérique est définie comme le produit du **nombre de pixels** que cette image a en largeur par le nombre de pixels de haut. La taille d'une image numérique, également connue sous le nom de poids de l'image, est souvent mesurée en octets, Mo ou Ko c-a-d Le nombre de lignes multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image [II .5]

II.4. 2 Résolution

La résolution d'une image est le nombre de pixels par pouce qu'elle contient (1 pouce = 2.54 centimètres). Elle est exprimée en "PPP" (points par pouce) ou DPI (dots per inch). Plus il y a de pixels (ou points) par pouce et plus il y aura d'information dans l'image (plus précise). [II .5]

II.4. 3 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.[II.6]



Fig. II. 3 Image sans bruit.

Image avec bruit.

II.4. 4 Histogramme

Il représente la distribution des intensités des pixels. Un histogramme est une fonction qui donne, pour chaque intensité lumineuse, le nombre de pixels ayant cette valeur. L'abscisse d'un histogramme $h(x)$ représente les niveaux d'intensité allant du plus foncé à gauche au plus clair à droite. Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse.

Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite). Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées. Prenons par exemple l'image suivante composée de niveaux de gris :

La figure (4) montre une image avec son histogramme. [II .4]

II.4. 5 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255.

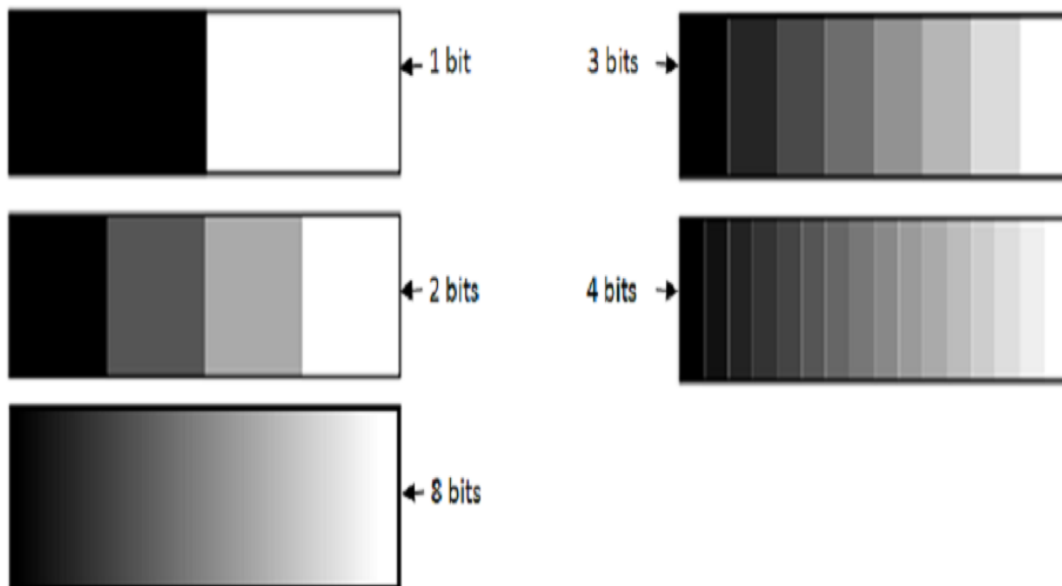


Fig. II. 4 Différent nuances avec différent nombres de bits. [II .7]



Fig. II. 5 Image au niveau de gris. [II .7]

II. 5 Types d'images

On distingue trois types d'images :

- Binaire : 2 couleurs (arrière-plan et avant-plan).
- Monochrome : variations d'une même teinte.
- Polychrome : " vraies " couleurs. [II .4]

M Sandeli, traitement d'images par des approches bio-inspirées application à la segmentation d'images, université Constantine 2. 2014

II. 5.1 Images binaires (en noir et blanc)

Les images binaires sont les plus simples. Bichromes (dont les pixels ne peuvent avoir que les valeurs 0 et 1) .Le 0 correspond à un pixel noir et 1 à un pixel blanc. Le niveau de gris est donc codé sur un seul bit.



Fig. II. 6 Image binaire

II. 5.2 Images à niveaux de gris (Monochromes)

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre **0 et 255**.

Chaque pixel n'est donc plus représenté par **1 bit**, mais par **1 octet**. Pour cela, il faut que le matériel utilisé pour afficher l'image, soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.[II.4]

M Sandeli, traitement d'images par des approches bio-inspirées application à la segmentation d'images, université constantine 2. 2014.



Fig. II. 7 Image Monochrome (256 couleurs).

II. 5.3 Images en couleurs (Polychromes)

Elle est obtenue par la combinaison de trois couleurs dites primaires : rouge, vert et bleu(RVB) .chaque couleur est codée comme une image à niveaux de gris, avec des valeurs allant de 0 à 255 .pour $R=V=B=0$ nous auront un noir pur, et pour $R=V=B=255$ nous auront un blanc pur. La représentation des images couleurs se fait donc soit par une image dont la valeur du pixel est une combinaison linéaire des valeurs des trois composantes couleurs, soit par trois images distinctes représentant chacune une composante couleur, on distingue généralement deux types d'images : [II .6]

- Images 24 bits ;
- Images à palettes ;



Fig. II. 8 Image polychrome (65536 couleurs).

II. 6 qualités de l'image numérique

Elle dépend, d'une part, de la qualité des images d'origine et, d'autre part, des moyens mis en œuvre pour convertir un signal analogique en signal numérique. Elle dépend aussi de :

- La qualité des périphériques de numérisation de l'image, du nombre de niveaux de gris ou de couleurs enregistrées, etc.
- La qualité de l'affichage à l'écran : définition de l'écran, nombre de teintes disponibles.[II.8]

Les critères d'appréciation de la qualité d'une image, tels que cités succinctement Ci-dessus, dépendent largement de la structure même de l'image réaliste ou conceptuelle et de son mode de représentation (bitmap ou vectorielle). [II .4]

II. 7 Système de traitement d'image

Un système de traitement d'image est généralement composé des unités suivantes :

- Un système d'acquisition et de numérisation qui permet d'effectuer l'échantillonnage et la quantification d'une image.
 - Une mémoire de masse pour stocker les images numérisées.
 - Un système de visualisation.

- Une unité centrale permettant d'effectuer les différentes opérations de traitement d'images.

[II.9]

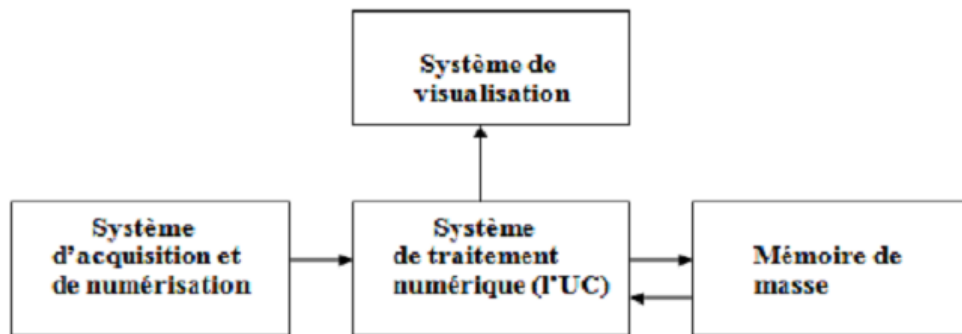


Fig. II. 9 Composition d'un système de traitement numérique.

II. 8 Les différents formats d'images

II. 8 .1. Images matricielles (Bitmap)

Image raster (ou bitmap) Se compose d'une grille de points ou de pixels. Tout le monde. Il peut être d'une couleur différente. En particulier, les images raster sont identifiées avec :

- Dimensions en pixels
- Dissolution
- Son mode de mesure des couleurs

Les images sur l'écran du téléviseur et les photos sont des images tramées.

Les images tramées sont également acquises à l'aide d'un appareil photo numérique.

Caméra vidéo numérique ou scanner. [7]



Fig. II. 10 Images matricielles [II .10]

II. 8 2. Images vectorielles

Les images vectorielles utilisent également la technologie des pixels, mais cette fois la position De plus, les couleurs ne sont pas fixes car elles sont calculées dynamiquement par le logiciel.

En d'autres termes, le logiciel détermine le point de départ, par exemple pour afficher une ligne. Point d'arrivée et trajectoire ultérieure. Puis calculez et placez tous les pixels Obligatoire pour afficher cette ligne. Il en va de même pour plus de formes et de couleurs. Compliqué.

Cette technique est souvent utilisée lors de l'utilisation ou de la création de palettes graphiques. Logos et dessins animés. [II .11]



Fig. II. 11 Images vectorielle [II .10]

II. 8 .3. JPEG

JPEG (Joint Photographic Experts Group) est une méthode de compression avec perte. Les images JPEG compressées sont généralement enregistrées au format de fichier JFIF (JPEG) Format de fichier d'échange). Il s'agit du format de fichier image le plus utilisé. Caméra Le numérique et les sites Web utilisent des fichiers JPG. Cependant, JPEG utilise la compression Il est irréversible et peut réduire considérablement la taille du fichier.

L'extension de fichier JPEG / JFIF est JPG ou JPEG. Presque tous les appareils photo numériques Les images peuvent être enregistrées au format JPEG / JFIF, qui encode les images en 8 bits Images en gris et en couleur 24 bits. [II .12]

II. 8 4. TIFF

Format TIFF (tagged image file format), utilisé pour stocker des images de haute qualité Noir et blanc, qualité des couleurs RVB jusqu'à 32 bits par pixel. Prend également en charge les images Il est indexé à l'aide de la palette de couleurs, des niveaux et des canaux alpha (transparence) [II .12,13].

II. 8 .5 GIF

GIF (Graphiques Inter change Format), un format léger, divertissement. Les images GIF ne peuvent contenir que 2, 4, 8, 16, 32, 64, 128 ou 256 couleurs 16,8 millions avec la palette de mode RVB. Les GIF sont limités aux palettes 8 bits ou 256 couleurs. [II .12,13]

II. 8 .6. PNG

Format de fichier PNG (Portable Network Graphics). Vous pouvez enregistrer l'image en noir. Blanc (jusqu'à 16 bits par pixel), couleurs vraies (jusqu'à 48 bits par pixel) Images indexées à l'aide de la palette de 256 couleurs. Enfin, il fournit Canal alpha à 256 niveaux pour la transparence.[II .12,13]

II. 9 Domaines d'application

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que les télécommunications (T.V., vidéo, publicité,...), la médecine (radiographie, ultrasons,...), biologie, astronomie, géologie, l'industrie (robotique, sécurité), la météorologie, l'architecture, l'imprimerie, l'armement (application militaire).

De nouvelles applications pratiques sont possibles aujourd'hui et touchent tous les domaines d'activités, tels que : métiers du spectacle, de la radio, créations artistiques,... [II .14]

II. 10 Conclusion

Nous avons présenté dans ce chapitre les notions de base qui servent de fondement à la compréhension des méthodes de traitement de l'images, ainsi les différents types d'images, une brève introduction aux concepts liés au domaine du traitement d'images ces notions qui sont utilisées dans la séparation aveugle source, particulièrement dans le chapitre quatre

CHAPITRE III
Algorithmes Evolutionnaires
et Optimisation

III.1 Introduction

Les algorithmes évolutionnaires (EA) sont un type d'intelligence artificielle. Ils sont motivés par des processus d'optimisation que nous observons dans la nature, tels que la sélection naturelle, la migration des espèces. Les chercheurs ont commencé à adopter les techniques de l'informatique évolutionnaire [III.1], qui utilisent les principes de l'évolution pour guider le processus d'optimisation. Il s'agit d'un sous-domaine de l'intelligence artificielle, plus précisément de l'intelligence computationnelle, qui implique des problèmes d'optimisation combinatoire [III.2].

Les techniques d'intelligence computationnelle utilisent des progrès itératifs [III.3], comme la croissance ou le développement d'une population. Cette population est ensuite sélectionnée dans une recherche aléatoire guidée utilisant un traitement parallèle pour atteindre l'objectif souhaité. Ces processus sont souvent inspirés des mécanismes biologiques de l'évolution. Habituellement, une première population de solutions générées aléatoirement comprend la première génération.

Une fonction objective est appliquée aux solutions et à tous les descendants suivants. Avec la sélection, les parents sont souvent choisis avec la valeur de fitness la plus élevée [III.4], pour la génération suivante.

Les parents se reproduisent grâce à l'application d'opérateurs tels que le croisement et la mutation. Le premier agit - souvent - sur les deux parents sélectionnés et les résultats sont d'un à deux enfants. Tandis que la mutation agit sur une solution et conduit à une nouvelle solution. Ces opérateurs créent une nouvelle population de solutions. Ce processus peut être répété jusqu'à ce qu'une population de solutions de qualité suffisante, ou un nombre de générations préalablement défini, soit atteint. L'optimisation évolutive présente un certain nombre d'avantages qui justifient les efforts investis dans ce domaine [III.5].

L'avantage le plus important réside dans le gain de flexibilité et d'adaptabilité de la tâche à accomplir, en combinaison avec des performances robustes et des caractéristiques de recherche globales [III.6].

Les techniques d'optimisation basées sur l'évolution utilisent (à chaque itération) une population de solutions au lieu d'une solution unique. Cela leur permet (en principe) d'identifier plusieurs solutions optimales dans leur population finale. Les techniques d'intelligence informatique proposent souvent des solutions bien adaptées à tous les types de problèmes, car elles ne font - théoriquement - aucune hypothèse sur le paysage de la fonction Fitness [III.3].

III.2 - Définition de l'optimisation :

L'optimisation est un acte ou une méthodologie pour rendre la conception, le système ou la décision parfaits et fonctionnels. Un problème d'optimisation est défini comme la recherche du minimum ou du maximum (optimal) pour une fonction donnée. On peut également trouver des problèmes d'optimisation dans lesquels les variables de la fonction à optimiser sont contraintes d'évoluer dans une partie spécifique de l'espace de recherche. Dans ce cas, nous avons une forme particulière de ce qu'on appelle un problème d'optimisation sous contrainte

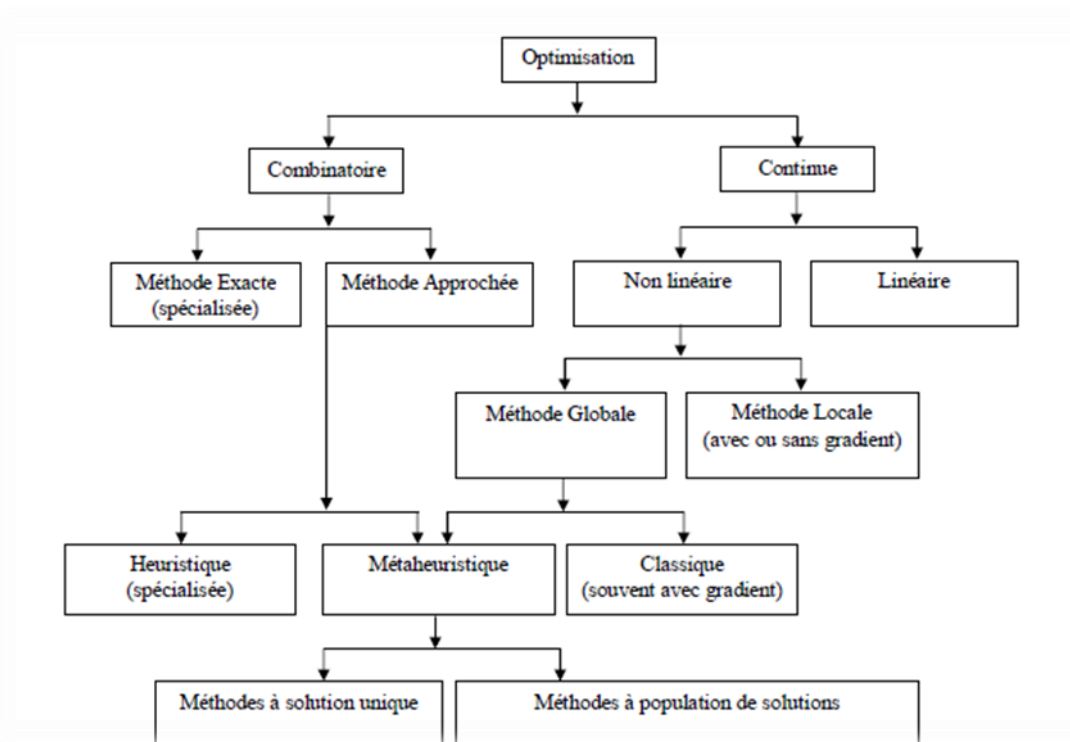


Fig. III. 1 Classification générale des méthodes d'optimisation

III.3 Les algorithmes évolutionnaires :

Les algorithmes évolutionnaires sont des méthodes d'optimisation stochastique basées sur une simulation de l'évolution naturelle d'une population. Plus précisément, il s'agit de techniques de programmation inspirées du principe d'évolution des espèces décrit par Darwin, ce qui explique comment les espèces depuis l'avènement de la vie ont pu évoluer de manière innovante. Flexible au sens de la meilleure adaptation au milieu, tout en ne permettant la reproduction que des individus bien adaptés à leur milieu [III.7].

En 1975, John Langton parle pour la première fois d'algorithmes génétiques (une sous-famille d'algorithmes évolutionnaires), une technique de programmation inspirée du principe d'évolution des espèces décrit par Darwin, mais ce n'est pas si récemment (vers le début des années 1990) que cette technologie fait son apparition. de la scène, enregistrant Un peu, avec les

technologies liées à la vie artificielle de manière plus générale, comme méthode de renouvellement des programmes de génie et de l'intelligence artificielle. On n'est guère surpris du succès de cette méthode de programmation quand on sait que la simplicité de cette technique n'a d'égal que sa puissance. C'est l'agilité de mise en œuvre qui fait toute la magie des algorithmes génétiques qui donne cette impression rafraîchissante d'avoir un peu de vie et de créativité dans l'ordinateur. Cet aspect à lui seul pourrait révolutionner la façon dont nous voyons les machines .qui nous entourentCe texte propose seulement, en attendant une couverture plus ambitieuse, l'introduction du principe initial qui est à la base de la programmation génétique [III.8]

III .3.1 Principe des algorithmes évolutionnaires :

Les phénomènes physiques ou biologiques ont été à l'origine de nombreux algorithmes d'inspiration assez libre. Ainsi, les réseaux de neurones artificiels s'inspirent du travail du cerveau humain, de l'algorithme de recuit simulé thermodynamiquement, et des algorithmes évolutifs (dont les AE) dont les plus connus sont algorithmes génétiques) pour l'évolution darwinienne des populations biologiques [III.9].

Les AE sont des techniques de recherche stochastiques qui ont été appliquées avec succès à un large éventail de problèmes réels et complexes. Un expert est une technique itérative qui utilise des facteurs de covariance aléatoires sur un « groupe » d'individus (population). Chaque membre de la communauté représente une version codée (solution) du problème. Initialement, cet ensemble initial est généré aléatoirement.

A chaque génération/itération de l'algorithme, des solutions sont sélectionnées, regroupées par paires et recombinaison afin de produire de nouvelles solutions qui remplacent les "pires" solutions selon un certain critère, etc.

La fonction d'évaluation associe la valeur de fitness à chaque individu afin de déterminer son adéquation au problème (critère de sélection) En ne permettant la reproduction que des individus bien adaptés à l'environnement, la nature assure la continuité de leurs meilleures caractéristiques, dont les caractéristiques se combinent entre elles (chaque enfant reçoit de bonnes caractéristiques à la fois de son père et de sa mère) pour former de nouveaux individus avec chaque génération, toujours mieux adaptés à leur environnement.

III .4 Définitions et notations :

« Soit pour optimiser une fonction à valeurs réelles F définie sur un espace W supposé de distance d .

- La fonction objectif F est appelée fonction de performance, ou fonction d'adaptation
- Les points de l'espace de recherche W sont appelés individus ;
- les P-uplets d'individus sont appelés populations ;
- On parlera d'une génération pour la boucle principale de l'algorithme", [III.9]

III .4.1 Le squelette :

« La pression de « l'environnement » est simulée à l'aide de la fonction d'adaptation F , et le principe darwinien « Les plus aptes survivent et se reproduisent », est implémenté comme suit :

1. Initialisation de la population P_0 en choisissant P individus dans W , généralement par tirage au sort avec une probabilité uniforme sur W ;
2. Evaluation des individus de P_0 (i.e. calcul des valeurs de F pour tous les individus)
3. La génération i construit la population P_i à partir de la population P_{i-1} :
 - Sélection des individus les plus efficaces (au sens de F) de P_{i-1} ;
 - Application (avec une probabilité donnée) d'opérateurs génétiques aux parents sélectionnés, ce qui génère de nouveaux individus, les enfants ; on parlera de mutation pour les opérateurs unaires, et de croisement pour les opérateurs binaires (ou n-aires) ;
 - Évaluation des enfants;
 - Remplacement des parents au moyen d'une sélection darwinienne parmi les enfants, avec participation éventuelle des parents.
4. L'évolution s'arrête lorsque le niveau de performance souhaité est atteint, ou qu'un nombre fixe de générations se sont écoulées sans améliorer l'individu le plus performant », [III.9].

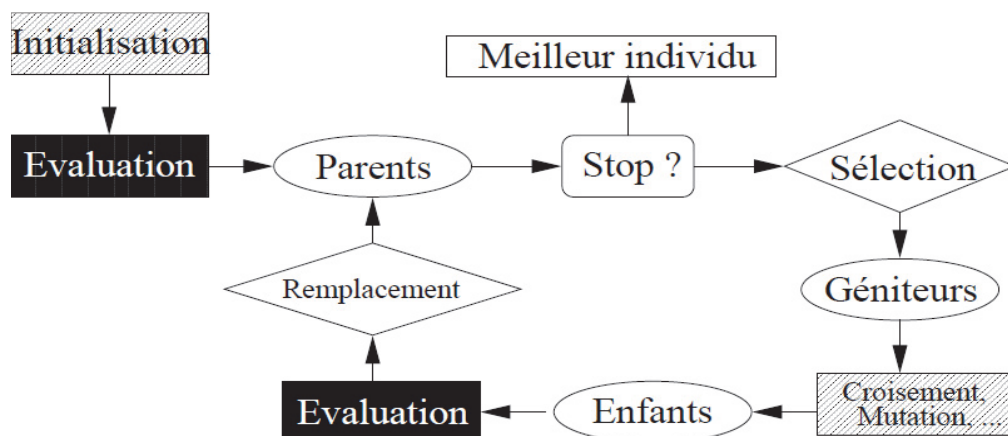


Fig. III. 2 Squelette d'un algorithme évolutionnaire [III.10].

III.5 La recherche tabou :

Dans la vie d'aujourd'hui, la technologie et l'information augmentent rapidement, ce qui rend difficile la recherche d'une solution optimale. Pour trouver une solution à un problème discret avec présence de contraintes ou de variables de décision, nous optons pour l'optimisation combinatoire. Afin de résoudre ce problème, le concept de méta-heuristique, appelé algorithme tabou (Tabu Search : TS), a été initialement introduit par Glover [III.11].

C'est l'une des heuristiques les plus efficaces pour trouver des solutions de qualité en un temps relativement court. La principale caractéristique de la recherche tabou repose sur l'utilisation d'un mécanisme qui s'inspire de la mémoire humaine [III.12].

Ces dernières années, plus d'une centaine d'articles présentant les applications de la recherche tabou ont été réalisés. Dans plusieurs cas, TS apporte des solutions très proches de l'optimalité et permet de résoudre les problèmes difficiles qui se posent. Ces succès ont rendu le TS extrêmement populaire pour ceux qui souhaitent trouver de bonnes solutions aux grands problèmes de combinaison rencontrés dans la pratique. Ainsi TS est une méthode mathématique d'optimisation, appartenant à la catégorie des techniques de recherche locale utilisées pour résoudre de très grands problèmes complexes [III.13], [III.14], [III.15].

TS améliore les performances d'une méthode de recherche locale. Cette technique utilise une ou plusieurs mémoires qui sont mises à jour et utilisées lors de la recherche. Une fois qu'une solution est déterminée, elle est marquée comme tabou. Pour examiner les régions de l'espace de recherche qui seraient laissées inexploitées par la procédure de recherche locale, le TS modifie la structure du voisinage de chaque solution au fur et à mesure que la recherche se poursuit.

Les solutions obtenues sont déterminées par l'utilisation de la structure de la mémoire. Lors de la recherche, ces solutions visitées sont mises dans une liste appelée liste tabou qui est une mémoire à court terme. Cette liste de longueur finie k stocke les k dernières solutions rencontrées.

III.5.1 Concepts de base de la recherche tabou :

- Utilisation d'une structure de mémoires souples pour mémoriser les régions visitées et inclure des mécanismes permettant d'interdire à l'algorithme certains mouvements pour ne pas retourner trop rapidement vers les régions visitées.
- Critère d'aspiration (souplesse du mécanisme de la liste tabou) : autorisation et acceptation de certains mouvements tabous.
- Stratégie d'intensification : exploitation plus approfondie des régions prometteuses visitées récemment et renforcement de la recherche dans ces régions.
- Stratégie de transformation : exploration des régions qui n'ont pas encore été visitées.

Algorithme de base de la recherche tabou

Tableaux. III. 1 Algorithme de la recherche tabou (TS).

Etape 1	Initialisation : générer une configuration initiale x_0 dans S . Poser $x=x_0$ et $f=f(x_0)$
Etape 2	$k=0$. Liste taboue $T=\emptyset$.
Etape 3	Répéter <ul style="list-style-type: none"> - Incrémenter $k=k+1$ et générer un sous-ensemble V de solutions de telle sorte que la condition taboue ne soit pas violée - Choisir (en tenant compte des restrictions tabou et du critère d'aspiration) x le meilleur élément dans V ($best(V(x))$, par rapport à la fonction f) <ul style="list-style-type: none"> - $x_{k+1} = best(V(x))$ - Si $(f(x_{k+1}) < f(x))$ alors $x = x_{k+1}$ - Mettre à jour la liste taboue T - Stop, si la condition d'arrêt est atteinte (éventuellement appliquer les stratégies d'intensification et de diversification). Sinon, aller à l'étape 3
Etape 4	- Retourner x la meilleure solution trouvée.

III.6. Algorithme d'optimisation basée sur la biogéographie :

L'algorithme basé sur la biogéographie (BBO : Biogeography-Based Optimization), développé par Dan Simon en 2008 [III.16], trouve ses origines dans la théorie de l'équilibre dynamique (également appelée théorie de la biogéographie insulaire), énoncée par MacArthur et Wilson [III.17].

La théorie de la biogéographie consiste en l'étude de la répartition spatiale des espèces vivantes (plantes et animaux) et des causes de cette répartition. Il traite de la manière dont la richesse spécifique (nombre d'espèces) est maintenue dans un système insulaire soumis à l'immigration et sur lequel les espèces disparaissent [III.18].

Elle stipule que les milieux insulaires sont à l'origine dépourvus d'espèces et que celles-ci y arrivent progressivement à partir de vastes régions (appelées « continents », bien qu'il ne s'agisse pas nécessairement de continents à proprement parler) ou d'îles voisines. Certaines espèces sont

aussi mieux équipées que d'autres pour conquérir de nouveaux territoires, elles ont donc une plus grande capacité à coloniser les milieux insulaires que d'autres. Les interactions compétitives sur l'île, en revanche, ont tendance à accélérer les extinctions. L'intersection de ces deux processus dynamiques permet d'expliquer la richesse actuelle du peuplement. A l'équilibre, il y a un renouvellement constant des espèces.

L'algorithme BBO manipule une population d'individus appelés îles (ou habitats). Chaque îlot représente une solution possible au problème à résoudre. La "fitness" de chaque île est déterminée par son HSI (Habitat Suitability Index), une mesure de la qualité d'une solution candidate, et chaque île est représentée par des SIV (Suitability Index Variables).

Une bonne solution au problème d'optimisation est une île avec un grand nombre d'espèces, ce qui correspond à une île avec un faible HSI.

Selon la théorie de MacArthur et Wilson, le nombre d'espèces présentes sur une île dépend essentiellement d'un équilibre entre le taux d'immigration de nouvelles espèces et le taux d'émigration des espèces déjà établies sur l'île. Dans BBO, chaque habitat a son propre taux d'immigration (λ) – arrivées de l'extérieur – et son propre taux d'émigration (μ) – départs vers l'extérieur. Ces paramètres sont influencés par le nombre d'espèce(s) présente(s) sur l'île.

Le taux d'immigration (λ) diminue avec l'augmentation du nombre d'espèce(s) déjà présentes sur l'île. Plus le nombre d'espèces déjà installées sur l'île augmente, moins il y a d'immigrants appartenant à une nouvelle espèce qui rejoignent l'île. Mais, à mesure que le nombre d'espèces déjà présentes sur l'île diminue, le taux d'immigration augmente. Le taux d'immigration maximal (I) est atteint lorsque l'île est vide. Une fois que toutes les espèces sont présentes sur l'île, alors $s = s_{max}$ (capacité maximale de l'île) et le taux d'immigration tombe à zéro, ne favorisant plus l'installation de nouveaux arrivants (plus l'île est peuplée, moins les espèces étrangères ont de chances de s'y installer). Le taux d'immigration, lorsqu'il y a s espèces sur l'île, est donné par :

$$\lambda = I \left(1 - \frac{s}{s_{max}} \right) \quad (\text{III.1})$$

Le taux d'émigration (μ) augmente avec le nombre d'espèces (s) présentes sur l'île. Le taux d'émigration maximum (E) se produit lorsque toutes les espèces sont présentes sur l'île ($s = s_{max}$), et devient nul si les espèces présentes sur l'île s'éteignent (ou quittent l'île). Le taux d'émigration quand il y a s espèces sur l'île est donné par :

$$\mu = E \left(\frac{s}{s_{max}} \right) \quad (III.2)$$

La figure III.3 représente graphiquement le modèle d'équilibre du nombre d'espèces sur les îles. Le nombre d'espèces déjà établies sur une île a un effet négatif sur l'immigration (compétiteurs, prédateurs et parasites déjà présents, moins d'espèces qui restent à immigrer), et un effet positif sur l'émigration (moins de ressources par espèce, forte compétition interspécifique).

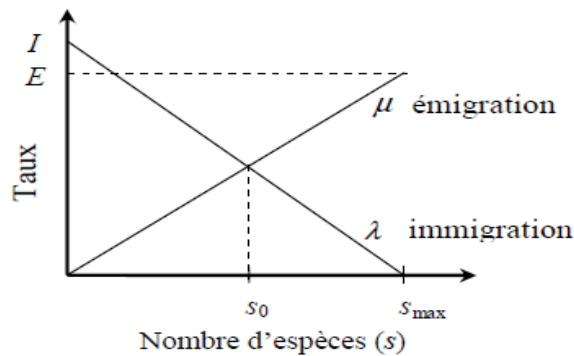


Fig. III. 3 Modèle linéaire de taux d'émigration (μ) et le taux d'immigration (λ) [III.16]

Le taux d'immigration chute rapidement au début lorsque les meilleurs colonisateurs s'établissent sur l'île. Le taux d'émigration s'accroît plus rapidement avec un nombre élevé d'espèces déjà présentes sur l'île. Le nombre d'espèces à l'équilibre sur l'île (s_0) est déterminé par l'intersection des courbes d'émigration (E) et d'immigration (I). Le modèle de la figure III.3 représente l'évolution du taux d'immigration (resp. d'émigration) par une fonction linéaire décroissante (resp. croissante) du nombre d'espèces présentes sur l'île. Il existe toutefois différents modèles mathématiques de la biogéographie, qui comprend des variables plus complexes [III.17].

Il y a, en effet, d'autres facteurs importants qui influencent les taux de migration entre les habitats, tels que la distance entre les habitats, la taille de l'habitat, variations climatiques (pluviométrie, température), la diversité végétale et animale, en plus de l'activité humaine. Ces facteurs rendent les courbes d'immigration et d'émigration plus complexes, contrairement à celles décrites dans le document original sur BBO [III.16].

Pour examiner l'influence de différents modèles de migration sur les performances de BBO, Haiping Ma [III.19] a étudié le comportement de six modèles de migration. Les résultats expérimentaux montrent clairement que les modèles de migration les plus proches de la nature (c'est-à-dire, non-linéaires) sont nettement mieux que les modèles linéaires.

Considérons à présent la probabilité P_i que l'île abrite exactement i espèces. Le nombre des espèces change pendant l'intervalle de temps Δt , $t \leq t + \Delta t < t + 2\Delta t$ selon l'équation suivante :

$$P_i(t + \Delta t) = P_i(t)(1 - \lambda_i \Delta t - \mu_i \Delta t) + P_{i-1} \lambda_{i-1} \Delta t + P_{i+1} \mu_{i+1} \Delta t \quad (III.3)$$

L'équation (III.3) stipule que le nombre des espèces sur l'île dépend du nombre total des espèces déjà établies sur l'île, de la fréquence à laquelle les nouvelles espèces arrivent et de la fréquence à laquelle les anciennes disparaissent. Nous supposons ici que Δt est assez petit pour que la probabilité que deux changements ou plus se produisent pendant un tel intervalle est nulle. Afin de disposer de i espèces à l'instant t , l'une des conditions suivantes doit être remplie :

- Il y a i espèces à l'instant t , et aucune immigration ni émigration n'a eu lieu entre l'instant t et l'instant $t + \Delta t$;
- Il y a $i - 1$ espèces sur l'île à l'instant t , et une nouvelle espèce s'y installe
- Il y a $i + 1$ espèces sur l'île à l'instant t , et une espèce quitte l'île.

La limite de (III.3) quand $\Delta t \rightarrow 0$ est donnée par l'équation (III.4).

$$P_i = \begin{cases} -(\lambda_i + \mu_i)P_i + \mu_{i+1}P_{i+1} & Si \quad i = 0 \\ -(\lambda_i + \mu_i)P_i + \lambda_{i-1}P_{i-1} + \mu_{i+1}P_{i+1} & Si \quad 1 \leq i \leq S_{max} - 1 \\ -(\lambda_i + \mu_i)P_i + \lambda_{i-1}P_{i-1} & Si \quad i = S_{max} \end{cases} \quad (III.4)$$

Où P_i est le changement de probabilité d'existence de l'habitat i .

L'algorithme BBO peut être universellement décrit par l'algorithme III.2. Les deux principaux facteurs qui régissent le fonctionnement de BBO sont la migration et la mutation. De plus

La stratégie de l'élitisme dans l'algorithme BBO, afin de maintenir la meilleure solution dans la nouvelle société

Tableaux. III. 2 Tableau III.2 : L'algorithme BBO

1. Générer aléatoirement un ensemble de solutions initiales (îles)
 2. **Tant que** le critère d'arrêt n'est pas atteint faire
 3. Évaluer la fitness (HSI) de chaque solution
 4. Calculer le nombre d'espèce s , le taux d'immigration λ et
d'émigration μ pour chaque solution
 5. **Migration :**
 6. pour $i = 1$ à NP faire
 7. | Utiliser λ_i pour décider, de manière probabiliste, d'immigrer à X_i
 8. | **si** $\text{rand}(0, 1) < \lambda_i$ **alors**
 9. | **pour** $j = 1$ à NP **faire**
 10. | Sélectionner l'île d'émigration X_j avec une probabilité $\alpha \mu_j$
 11. | **si** $\text{rand}(0, 1) < \mu_j$ **alors**
 12. | **Remplacer une variable de décision (SIV) choisie**
aléatoirement dans X_i par la variable correspondante dans X_j
 13. | **fin**
 14. | **fin**
 15. | **fin**
 16. | **fin**
 17. **Mutation :** Muter les individus au taux de mutation donné par l'équation (III.5).
 18. Remplacement de la population par les descendants
 19. Implémenter l'élitisme
 20. **fin**
-
- 21 retourner la meilleure solution trouvée

L'idée générale de la migration est l'échange de caractéristiques entre les îles. Les taux d'immigration (λ) et d'émigration (μ) de chaque île sont utilisés pour transmettre, de manière probabiliste, les caractéristiques entre les îles, $\text{rand}(0, 1)$ est un nombre aléatoire uniformément distribué dans l'intervalle $[0, 1]$ et $X_{i,j}$ est le $J^{\text{ème}}$ SIV de la solution X_i .

La stratégie de migration de BBO est similaire à la recombinaison des stratégies d'évolution [III.20], dans laquelle plusieurs parents sont recombinaison entre eux pour former un unique enfant. La principale différence réside dans le fait que la recombinaison est utilisée pour créer de nouvelles solutions, tandis que la migration est utilisée pour modifier des solutions existantes.

Le HSI d'une île peut changer brusquement, en raison d'événements aléatoires : des catastrophes naturelles (tempêtes, ouragans, incendies, etc.) ou des épidémies, etc. BBO modélise ce phénomène comme une mutation des SIVs, et utilise les probabilités de nombre d'espèces (species count probabilités P_i) pour déterminer les taux de mutation. La mutation est utilisée pour améliorer la diversité de la population, empêchant ainsi la recherche de stagner. La probabilité, qu'une solution donnée i , existe a priori comme une solution pour le problème considéré, est spécifiée par la probabilité du nombre d'espèces (P_i). Si une île i est sélectionnée pour la mutation, alors une variable SIV est modifiée de façon aléatoire en fonction de sa probabilité P_i . Dans ce contexte, il convient de remarquer que des solutions avec des valeurs de HSI très élevées ou très faibles ont une faible probabilité d'exister. Tandis que les solutions avec un HSI moyen sont relativement probables. Si une solution donnée a une probabilité faible, elle est susceptible d'être mutée à une autre solution. A l'inverse, une solution avec une forte probabilité est en revanche moins susceptible d'être mutée.

Le taux de mutation m_i pour l'habitat i est calculé en fonction de la probabilité de la solution exprimée dans l'équation (III.5).

$$m_i = m_{max} \left(\frac{1-P_i}{P_{max}} \right) \quad \text{(III.5)}$$

où m_{max} est le taux maximum de mutation, et $P_{max} = \operatorname{argmax} P_i, i=1, \dots, s_{max}$. Si une île est sélectionnée pour la mutation, alors un SIV choisi au hasard dans l'île est simplement remplacé par une variable aléatoire générée dans son intervalle de définition.

III.7 L'algorithme d'ascension de collines (Hill Climbing) :

Hill Climbing tire son nom du fait que l'espace fonctionnel est supposé être "en forme de colline". C'est une technique d'intelligence artificielle qui vise à trouver numériquement des solutions à l'équation diophantienne (une équation diophantienne est une équation polynomiale à une ou plusieurs inconnues dont les solutions sont recherchées parmi des nombres entiers, éventuellement rationnels, les coefficients étant eux aussi entiers). Ces équations sont importantes car elles ont de nombreuses applications dans des domaines tels que la cryptographie à clé

publique, la factorisation d'entiers, les courbes algébriques, les courbes projectives et la dépendance des données dans les supercalculateurs.

Un fait important : il a été prouvé qu'il n'existe pas de méthode générale pour trouver des solutions à de telles équations. La méthode qui utilise une représentation arborescente pour décrire les solutions possibles des équations diophantiennes adopte une nouvelle méthodologie pour générer des successeurs. La fonction heuristique utilisée permet de faire du processus de recherche de solutions un processus de minimisation [III.21].

L'algorithme Hill-Climbing est une implémentation de base de la recherche locale. A partir d'un candidat initial généré aléatoirement, l'algorithme remplace itérativement le candidat actuel par un meilleur, en cherchant parmi les alternatives locales un voisin.

Dans l'algorithme, la solution s garde la trace du meilleur candidat, à tout moment. Si le score de s atteint 0, ce qui signifie qu'il s'agit d'un optimum global, l'algorithme s'arrête. En revanche, si chaque voisin de x a été évalué sans remplacer s comme candidat amélioré, l'algorithme s'arrête sans indication globale des optima. Lorsqu'il atteint un candidat dont le voisinage ne contient aucune solution améliorée, il renvoie la solution courante.

Le voisinage d'un candidat actuel est généralement défini comme un ensemble contenant tous les candidats possibles qui diffèrent du candidat actuel avec une petite modification. La modification dépend de la structure des candidats. Pour les vecteurs, le voisinage peut être défini comme tous les vecteurs où une valeur est incrémentée ou décrémentée de un par rapport au candidat. Cette technique se distingue des méthodes de descente traditionnelles en acceptant également des solutions de qualité égale. Ainsi, cette méthode choisit le premier voisin qui a une qualité meilleure ou équivalente.

L'algorithme de base de Hill Climbing est le suivant :

Tableaux. III. 3 . L'Algorithme du Hill Climbing (HC).

Entrée : Une solution candidate x

Sortie : Solution s ou nul si impossible de trouver une solution valide

1 : $s \leftarrow x$

2 : Tant que $\text{score}(s) > 0$ faire

3 : pour chaque $x \in N(x)$ faire

4 : $s \leftarrow \min((x), (s))$

5 : Fin pour

```

6 : Si  $s=x$  Alors
7:   Return Null
8: Sinon
9:    $x \leftarrow s$ 
10: End if
11 : Fin Tant que
12 : Résultat  $\leftarrow s$ 

```

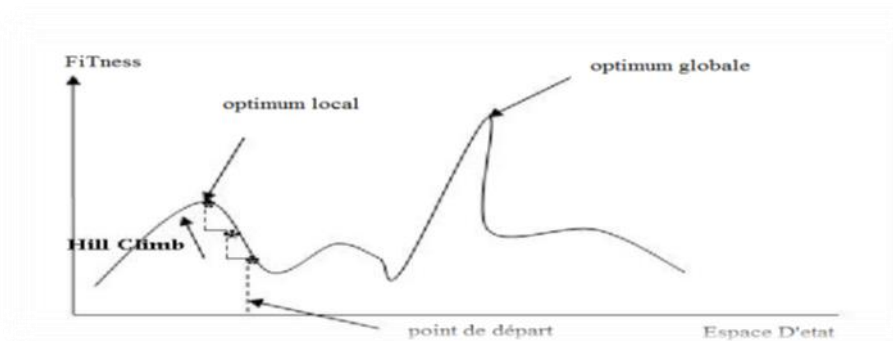


Fig. III. 4 Schéma de principe de base de l'algorithme Hill Climbing (HC)

Il y a quelques inconvénients avec cet algorithme métaheuristique. L'algorithme Hill Climbing est idéal pour trouver rapidement des optima locaux, mais échoue souvent à trouver la meilleure solution globale [III.22].

En effet, il est très vulnérable aux solutions et plateaux optima locaux. Comme son nom l'indique, la procédure monte continuellement au sommet d'une colline en faisant un petit pas à la fois. Lorsqu'elle n'a pas de prochaine étape qui puisse la mener au sommet, elle se considère comme étant au sommet.

Le problème est qu'il ne voit qu'une étape à la fois en termes d'amélioration du candidat. La figure III.4 illustre ces défis. Étant donné que l'optimum global est le sommet le plus élevé, un optimum local est un candidat avec un score inférieur à l'optimum global, mais tous les candidats dans le score de voisinage de l'optimum local sont encore plus bas, ce qui entraîne l'arrêt de l'algorithme Hill Climbing.

Un plateau est une situation où, pour un certain nombre d'échelons à partir des quartiers des candidats, on choisit le meilleur candidat à chaque échelon qui aboutit au même résultat. Lorsque l'algorithme pense avoir atteint le sommet, cela pourrait en fait être un moyen de continuer à chercher une meilleure solution s'il est prêt à poursuivre certaines étapes

intermédiaires sans faire de progrès. Compter plus d'étapes, à chaque itération, est une possibilité, mais cela augmente la complexité de la procédure.

Le seul avantage du Hill Climbing est que la recherche d'une solution peut se dérouler en plusieurs étapes et rend ce temps métaheuristique plus efficace. On peut donc améliorer la solution sans trop augmenter la complexité, en exécutant plusieurs fois l'algorithme Hill Climbing avec des candidats initiaux différents.

Dans cette approche, l'algorithme Hill Climbing s'exécute un certain nombre de fois ou jusqu'à ce qu'une limite de temps soit atteinte. En générant aléatoirement un candidat initial à chaque fois, cela fournit différents candidats optimaux locaux parmi lesquels l'algorithme choisit la meilleure solution, à moins que l'une des exécutions n'aboutisse à un optimum global, qui est immédiatement renvoyé.

III.8 L'algorithme du Recuit Simulé :

Le recuit simulé (SA) a été inventé par S. Kirkpatrick, CD. Gelatt et MP Vecchi en 1983[III.23].

Ce nom vient du recuit en métallurgie, une technique impliquant le chauffage et le refroidissement contrôlés d'un matériau afin d'améliorer ses propriétés structurales en éliminant les défauts du cristal. Le recuit simulé est une méthode pour résoudre des problèmes d'optimisation sans contraintes ou contraintes liées.

La méthode modélise le processus physique de chauffage d'un matériau, puis d'abaissement lent de la température pour réduire les défauts, minimisant ainsi l'énergie du système. Le recuit simulé appartient à une classe d'algorithmes de recherche locale qui suivent une stratégie stochastique. Ces algorithmes jouent un rôle particulier dans la recherche locale pour deux raisons. Premièrement, ils semblent être très efficaces lorsqu'ils sont appliqués à un large éventail de problèmes pratiques [III.23].

La méthode d'optimisation SA provient d'un schéma d'échantillonnage conçu par Metropolis et al. [III.24] d'effectuer une intégration de Monte Carlo de plusieurs centaines d'intégrales dimensionnelles nécessaires pour évaluer les propriétés d'équilibre d'un ensemble de molécules à une température T donnée.

La méthode SA peut être classée en trois catégories, recuit simulé CSA classique (SA classique), le recuit simulé rapide FSA (Fast SA) et le recuit simulé généralisé GSA (Generalized SA).

Le recuit simulé FSA est une recherche semi-locale et consiste en de longs sauts occasionnels. L'algorithme FSA est inversement linéaire dans le temps et est rapide par rapport au recuit simulé classique (CSA) dont les fonctions de coût de processus possèdent des degrés assez arbitraires de non-linéarités, de discontinuités et de caractère aléatoire.

Il s'agit de conditions aux limites assez arbitraires et de contraintes imposées à ces fonctions de coût qui s'implémentent assez facilement avec un degré de codage assez minimal par rapport aux autres algorithmes d'optimisation non linéaire, ce qui garantit statistiquement qu'une solution sera trouvée. optimal [III.25].

L'algorithme GSA ou machine de Tsallis est un processus markovien stochastique, basé sur les statistiques de Tsallis qui a été utilisé dans de nombreuses classes de problèmes, comme la physique et la chimie et en particulier dans l'optimisation des systèmes moléculaires et les problèmes de repliement des protéines [III.26] [III.27].

La famille d'algorithmes de recuit simulé dépend d'une fonction de visite qui détermine comment le domaine de la fonction est recherché, et d'une fonction d'acceptation qui indique si un résultat d'énergie plus élevée doit être accepté ou rejeté. Dans l'algorithme Metropolis, on part d'une forme donnée et on soumet le système à une modification élémentaire. Si cette perturbation a pour effet de réduire la fonction objective (ou l'énergie) du système, elle est acceptée. Sinon, il est accepté avec probabilité P .

En répétant ce processus, on génère une suite de formes qui tendent vers l'équilibre thermodynamique. L'algorithme de recuit simulé est différent de l'algorithme Hill Climbing dans la décision de remplacer la solution S , par une meilleure solution. Plus précisément dans l'algorithme de recuit simulé, si R est meilleur que S , nous remplaçons la solution S par R comme d'habitude. Mais si R est pire que S , nous pouvons toujours remplacer S par R avec probabilité, $(t,.)$, avec L'algorithme du Recuit Simulé est comme suit :

$$P(t, R, S) = e^{(f(R) - \frac{f(S)}{t})} \quad (\text{III.6})$$

Tableaux. III. 4 L'algorithme d'optimisation Recuit Simulé

- | |
|---|
| <ol style="list-style-type: none"> 1: $t \leftarrow t_0$; t_0 est la température initialisée à une grande valeur 2: $s \leftarrow s_0$; s_0 une solution initiale 3: $best \leftarrow s$ 4: Tant que (best est la meilleure solution) & (Number-Iter =max) & ($t \leq 0$) Faire 5: $r \leftarrow s'$ où $s' \in (s)$ |
|---|

```

6:  $\rho \leftarrow \text{random}(0,1)$  ;
7: Si ( $f(s) < f(\text{best})$ ) ou ( $\rho < e^{(f(r)-f(s))/t}$ ) alors
8:  $s \leftarrow r$ ;
9: Si ( $f(s) - f(\text{best}) > 0$ )
10:  $\text{best} \leftarrow s$ ;
11: décrémenter  $t$  ;
12: Fin Tant que

```

III.8.1 Interprétation

Par exemple, nous avons un paramètre de température t , que nous voulons ajuster. Si la température t est proche de 0, la fraction $(f(r)-f(s))/t$ devient un grand nombre, alors la probabilité est proche de 0. Si t est très grand, la probabilité est proche de 1.

L'idée est d'abord initialiser la température à un grand nombre, ce qui donne à l'algorithme une exploration très aléatoire de l'espace des solutions. Diminuez ensuite la température t de manière lente, jusqu'à ce que la température converge vers zéro.

Dans ce cas, l'algorithme de recuit simulé rejoint l'algorithme de base de Hill Climbing.

III.9 Algorithmes génétiques :

John Holland, ses collègues et ses étudiants ont développé les Algorithmes Génétiques (AG) [III.28] à l'Université du Michigan, des métaphores biologiques inspirées des mécanismes de l'évolution darwinienne (sélection naturelle) et de la génétique.

Ces métaphores prennent la forme d'algorithmes de recherche appelés « algorithmes génétiques » [III.29]

Ces algorithmes appartiennent à la classe des algorithmes dits stochastiques.

En effet une grande partie de leur fonctionnement repose sur le hasard. Bien qu'ils utilisent le hasard, les AG ne sont pas purement aléatoires. Ils exploitent efficacement les informations obtenues précédemment pour spéculer sur la position de nouveaux points à explorer, dans l'espoir d'améliorer les performances [III.30], [III.28]

Les algorithmes génétiques permettent à une population de solutions de converger vers les solutions optimales. Pour ce faire, ils utiliseront un mécanisme de sélection des individus dans la population (les solutions potentielles). Les individus sélectionnés seront croisés entre eux (exploitation), et certains seront mutés (exploration).

Ces mécanismes d'exploitation et d'exploration permettront de converger vers les bonnes solutions en évitant au maximum les optima locaux [III.30]

III.9.1 Terminologie et éléments de base :

Un algorithme génétique recherche les extrêmes d'une fonction définie sur un espace de données appelé population. Par analogie avec la génétique, chaque individu de cette population est un chromosome et chaque caractéristique de l'individu est un gène.

Dans un cas simple, un gène sera représenté par un bit (0 ou 1), un chromosome par une suite de bits. Chaque gène représente une partie élémentaire du problème, il peut être assimilé à une variable et peut prendre différentes valeurs appelées allèles. La position du gène dans le chromosome est appelée locus [III.31].

Aussi appelé génotype et phénotype. Le génotype représente toutes les valeurs des gènes du chromosome tandis que le phénotype représente la vraie solution après transformation du chromosome.

Lors de la génération d'une nouvelle population, des opérateurs génétiques tels que la sélection, le croisement et la mutation sont nécessaires pour la manipulation des chromosomes [III.31].

Le tableau 2.1 présente un résumé de la terminologie naturelle et celle utilisée par les algorithmes génétiques [III.28], [III.30].

Tableaux. III5 . Comparaison de la terminologie naturelle et celle des algorithmes génétiques

Nature	Algorithme génétique
Chromosome	Chaîne
Gène	Trait, caractéristique
Allèle	Valeur de la caractéristique

Locus	Position dans la chaîne
Génotype	Structure Ensemble des valeurs des gènes
Phénotype	Ensemble de paramètres, structure décodée Evaluation d'un génotype

Les GA utilisent donc un vocabulaire proche de celui de la génétique. On parlera alors d'individus ou de chromosomes dans une population. Chaque individu ou chromosome est constitué d'un ensemble d'éléments appelés gènes contenant les caractéristiques héréditaires de l'individu [III.29].

Ils utilisent un mécanisme de sélection naturelle, basé essentiellement sur la reproduction et sur un codage génétique qui stocke l'information décrivant l'individu sous forme de gènes [III.32] imitant les systèmes naturels d'évolution des espèces

III.9.2 Évolution des espèces

Dans tout milieu dans lequel vit une population primitive, c'est-à-dire mal adaptée à ce milieu. Bien sûr, bien que globalement inadaptée, cette population n'est pas homogène : certains individus sont mieux armés que d'autres pour profiter des ressources offertes par l'environnement (nourriture, abri, etc.) et faire face aux dangers qui s'y cachent. (Prédateurs, intempéries, etc.).

Ces individus mieux équipés ont donc une plus grande probabilité de survie que leurs congénères et auront de fait d'autant plus de chances de pouvoir se reproduire.

En se reproduisant entre individus bien adaptés, ils transmettront à leurs enfants ces caractéristiques qui les ont rendus excellents. La population qui résultera de cette reproduction sera donc globalement mieux adaptée au milieu que la précédente puisque la plupart des individus auront hérité de plusieurs (puisque chacun hérite à la fois de leur mère et de leur père) des caractéristiques de « l'élite ». » De la génération précédente.

Et c'est ainsi, en recombinaison à chaque génération les caractéristiques élémentaires d'une bonne adaptation et en saupoudrant le tout d'un peu de chance, que la population évoluera vers une toujours mieux en adéquation avec l'environnement.

Par analogie, les AG rejoignent le même principe, ils reposent sur le principe « d'évolution » d'une population d'individus. Dans celui-ci, ce sont généralement les plus forts, c'est-à-dire les mieux adaptés au milieu, qui survivent et engendrent une progéniture. Basé sur les données de problème, nous créons (généralement aléatoirement) une « Population » de solutions admissibles. Ensuite, nous évaluons chacune des solutions. On élimine une infime partie de ceux qui se sont révélés inutiles, et on recombine les gènes des autres pour obtenir de nouvelles solutions individuelles.

Ainsi, à chaque génération, un nouvel ensemble de constructions (chaînes) est créé en utilisant des parties des meilleurs individus de la génération précédente ainsi que des parties innovantes. Selon théorie de l'évolution, cette nouvelle génération sera globalement plus adaptée au problème que la précédente. Ce processus est ensuite répété jusqu'à la naissance d'une solution qui sera jugée satisfaisante [III.31].

III.9.3 Conception d'un algorithme génétique

La simplicité de mise en œuvre et l'efficacité sont deux des caractéristiques les plus intéressantes de l'approche proposée par les GA. L'implémentation d'un algorithme génétique nécessite la disponibilité [III.28], [III.31] :

- Une représentation génétique du problème, c'est-à-dire un codage approprié des solutions sous forme de chromosomes. Cette étape associe chaque point de l'espace de recherche à une structure de données. Elle est généralement placée après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques ;
- Un mécanisme de génération de la population initiale. Ce mécanisme doit pouvoir produire une population inhomogène qui servira de base aux générations futures. Le choix de la population initiale est important car elle peut prendre plus ou moins rapidement la convergence vers l'optimum global. Dans le cas où rien n'est connu sur le problème à résoudre, il est indispensable que la population initiale soit répartie sur l'ensemble du domaine de recherche ;
- une fonction d'évaluation pour mesurer la force de chaque chromosome ;

- une méthode de sélection des chromosomes à reproduire ;
- des opérateurs permettant de diversifier la population au fil des générations et d'explorer l'espace de recherche. L'opérateur de croisement recompose les gènes des individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche ;
- des valeurs pour les paramètres utilisés par l'algorithme : taille de la population, nombre total de générations ou critère d'arrêt, probabilités de croisement et de mutation.

III.9.4 Comment fonctionne l'algorithme génétique ?

Un algorithme génétique fonctionne généralement selon un cycle simple de quatre étapes [III.31] :

1. création d'une population de chromosomes ;
2. évaluation de chaque chromosome ;
3. sélection des meilleurs chromosomes ;
4. la manipulation génétique, pour créer une nouvelle population de chromosomes

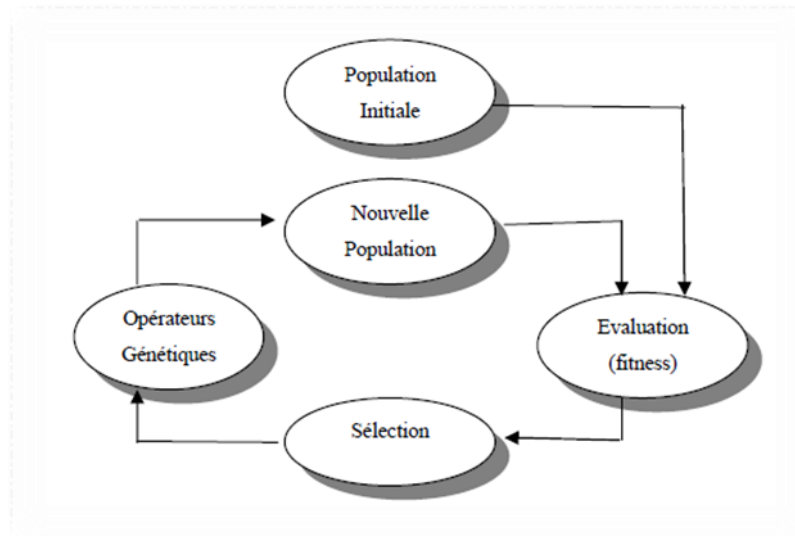


Fig. III. 5 Cycle génétique

Le cycle décrit à la figure III.5 s'inspire de la terminologie génétique. Au cours de chaque cycle, une nouvelle génération de solutions de problèmes est obtenue.

Initialement, une population initiale est générée dans laquelle chaque solution individuelle de la population est codée sous la forme d'une chaîne de caractères (chromosomes). Ensuite, une

évaluation de chaque chromosome sera établie. Cette évaluation consiste à évaluer la Qualité des chromosomes à l'aide de la fonction d'évaluation : fitness. Cela permet de sélectionner les chromosomes les plus adaptés et par conséquent de leur appliquer des opérateurs génétiques (croisement et mutation), ce qui crée une nouvelle génération.

A la fin du cycle, une nouvelle population est acquise ouvrant ainsi la voie à une nouvelle génération et par conséquent à un nouveau cycle.

III.9.5 Variantes

En fait, les algorithmes génétiques sont une famille d'algorithmes, basés autour des mêmes idées. Cependant, il existe de nombreuses variantes possibles selon la représentation choisie, les opérateurs de croisement, de mutation et de sélection [III.32]. La section suivante présente les choix les plus courants qui définissent les variantes.

III.9.5.1 Codage

Le codage est une modélisation d'une solution à un problème donné sous la forme d'une séquence de caractères appelée chromosome où chaque caractère, également appelé gène, représente une variable ou une partie du problème. La tâche principale est de choisir le contenu des gènes qui facilite la description du problème et respecte ses contraintes [III.30]. La littérature définit deux types de codage : binaire et réel.

III.9.5.1.a Codage binaire

Le codage classique utilise l'alphabet binaire : 0.1. Dans ce cas, le chromosome représente simplement une série de 0 et de 1. Le codage binaire est également indépendant des opérateurs génétiques (croisement et mutation) tant que ceux-ci ne nécessitent aucune spécification. En effet, toute manipulation d'un chromosome donne naissance à un nouveau chromosome valide. En pratique, le codage binaire peut présenter des difficultés. En effet, il est parfois très difficile ou très lourd de coder des solutions de cette manière. De plus, dans certains cas la taille mémoire requise peut devenir prohibitive [III.30].

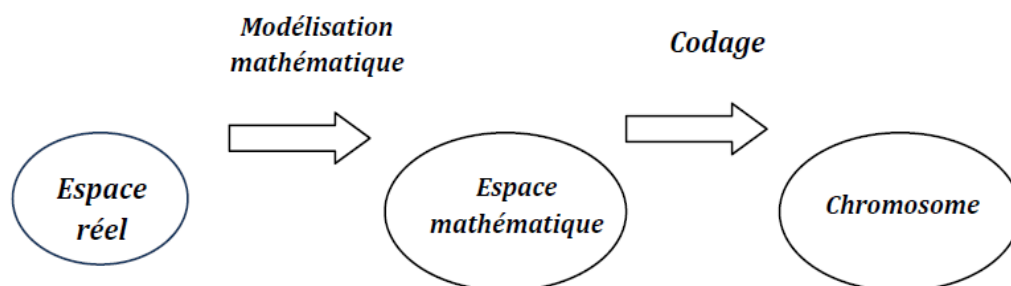


Fig. III. 6 Les phases de définition d'un codage

III.9.5.1.b Codage réel

Pour certains problèmes d'optimisation, il est plus pratique d'utiliser un codage réel des chromosomes. Un gène est ainsi représenté par un nombre réel au lieu de devoir coder les nombres réels en binaire puis les décoder pour les transformer en solutions efficaces. Le codage réel augmente l'efficacité de l'algorithme génétique e pour éviter des opérations de décodage supplémentaires. En effet, un chromosome codé en réels est plus court qu'un chromosome codé en binaire.

III.9.5.2 Évaluation : condition physique (fitness)

L'opérateur d'évaluation n'est pas trivial. Il est utilisé par l'opérateur de sélection pour choisir les individus à conserver. Ainsi, pour mesurer la performance de chaque individu qui correspond à une solution donnée du problème à résoudre, une fonction d'évaluation est introduite.

Elle permet de quantifier la capacité d'un individu à survivre en lui attribuant un poids communément appelé fitness. La force de chaque chromosome de la population est calculée pour que les plus forts soient conservés (étape de sélection) puis modifiés (crossover et mutation). La complexité de la fonction d'évaluation dépend essentiellement du problème et de ses contraintes [III.31], [III.30].

Ces deux derniers éléments, codage et évaluation, sont les seuls éléments spécifiques au problème à résoudre. Une fois fixés, l'algorithme génétique qui sera appliqué sera toujours le même [III.33].

III.9.5.3 Population initiale

Une fois le codage choisi, une population initiale constituée des solutions admissibles du problème doit être déterminée. Plusieurs mécanismes de génération de la population initiale sont utilisés dans la littérature [III.31].

Le choix de l'initialisation se fera en fonction de la connaissance du problème par l'utilisateur. S'il ne possède pas d'information particulière, alors une initialisation aléatoire, aussi uniforme que possible afin de favoriser une exploration de l'espace de recherche maximum, sera la plus adaptée. Mais en autres cas, il est possible d'utiliser d'autres mécanismes.

De plus, cette étape présente un problème principal qui est celui du choix de la taille de la population. En effet, une population trop importante augmente le temps de calcul et nécessite un espace mémoire important, tandis qu'une population trop petite conduit à l'obtention d'un optimum local.

III.9.5.4 Critère d'arrêt

Déterminer l'arrêt d'un processus génétique est l'une des difficultés majeures de l'approche génétique. En effet, si l'on excepte le cas des problèmes artificiels, on ne sait jamais si on a trouvé l'optimum. En pratique, l'utilisateur déclare un certain nombre de générations maximales.

La recherche peut également être arrêtée lorsque tous les individus d'une même population sont des copies d'un même individu. On dit alors qu'il y a « perte de diversité génétique » [III.34].

Les critères d'arrêt peuvent alors se résumer ainsi :

1. Arrêt après un nombre de générations fixé a priori.
2. Arrêt lorsque la population cesse de changer ou lorsqu'une population homogène est présente.

III.9.5.5 Sélection

L'opérateur de sélection est chargé de « favoriser » les meilleurs individus [III.32]. Plus formellement, l'opérateur de sélection va générer à partir de la population courante une nouvelle population en copiant les individus sélectionnés de la population courante. Les chaînes sont copiées en fonction des valeurs de la fonction d'adaptation. Ce processus donne aux meilleurs canaux une forte probabilité de contribuer à la prochaine génération. Cet opérateur est bien sûr une version artificielle de la sélection naturelle, la survie darwinienne des chaînes les mieux ajustées [III.28].

Il existe de nombreuses techniques de sélection, les plus courantes seront abordées dans la section suivante.

◆ Sélection par classement : elle consiste à ranger les individus de la population par ordre croissant (ou décroissant selon l'objectif) et à retenir un nombre fixe de génotypes. Ainsi, seuls les individus les plus forts sont retenus. L'inconvénient majeur de cette méthode est la convergence prématurée de l'algorithme génétique. Il est parfois nécessaire de garder quelques individus jugés faibles pour créer de la diversité au niveau de la population.

UN autre difficulté consiste à fixer une limite à la sélection, ce qui empêche parfois de conserver les bons candidats pour les générations futures [III.31].

◆ Sélection par roulette : elle consiste à créer une roue de loterie biaisée pour laquelle chaque individu de la population occupe une section de la roue proportionnelle à sa valeur d'évaluation. Ainsi, même les individus les plus faibles ont une chance de survivre. Si la population d'individus est de taille égale à N , alors la probabilité de sélection d'un individu x_i notée $p(x_i)$ est égale à :

$$P(x_i) = \frac{F(x_i)}{\sum_{k=1}^N F(x_k)} \quad (\text{III.7})$$

En pratique, on calcule pour chaque individu sa probabilité cumulée et on choisit au hasard un nombre r compris entre 0 et 1.

$$q_i = \sum_{j=1}^i P(x_j) \quad (\text{III.8})$$

L'individu sélectionné est x_i Si $q_1 \geq r$ Ou $x_i (2 \leq i \leq N$ Si $q_{i-1} < r \leq q_i$. Ce processus est répété N fois. Avec une telle sélection, un individu fort peut être choisi plusieurs fois. En revanche, un individu faible a moins de chances d'être sélectionné [III.31].

C'est cette sélection qui a été exclusivement utilisée dans cette thèse.

♦ Sélection du tournoi : elle consiste à choisir au hasard deux ou plusieurs individus et à sélectionner le plus fort. Ce processus est répété plusieurs fois jusqu'à l'obtention de N individus. L'avantage d'une telle sélection est d'éviter qu'un individu très fort soit sélectionné plusieurs fois [III.31].

III.9.5.6 Croisement

La naissance d'un nouvel individu nécessite le prélèvement aléatoire d'une partie des gènes de chacun des deux parents. Ce phénomène, qui vient de la nature, est appelé croisement.

C'est un processus essentiel pour explorer l'espace des solutions possibles. Une fois la sélection terminée, les individus sont répartis au hasard par paires. Les chromosomes parents sont ensuite copiés et recombinaison pour produire chacun deux descendants avec des caractéristiques des deux parents.

Afin de conserver quelques individus parents dans la population suivante, l'algorithme génétique est associé à une probabilité de croisement, qui permet de décider si les parents seront croisés entre eux ou s'ils seront simplement recopiés dans la population suivante.

[III.31], [III.32].

La littérature définit plusieurs opérateurs de croisement. Ils diffèrent selon le type de codage adapté et la nature du problème traité.

III.9.5.6.1 Croisement binaire

Ce croisement peut avoir recours à plusieurs types en occurrence [III.31]

♦ *Croisement en 1-point :*

Parent1 :	0	1	1	0	1	1	0	1
Parent2 :	1	1	0	0	1	0	0	1
Fils 1 :	0	1	1	0	1	0	0	1
Fils 2 :	1	1	0	0	1	1	0	1

Fig. III. 7 Croisement en un point de deux chromosomes

♦ *Croisement en n-points :* Ce type de croisement se traduit par un choix aléatoire de n-points de coupure pour dissocier chaque parent en n+1 fragments. Pour former un enfant, il suffit de concaténer alternativement n+1 sous-chaînes des deux parents. Ce croisement cherche à explorer tout l'espace des solutions possibles en créant une progéniture aux caractéristiques très éloignées des parents.

♦ *Croisement à 2 points :* c'est un cas particulier de croisement à n points. Deux points de coupure sont choisis au hasard pour créer les descendants.

Parent1	1 1 0 1	0 0 0 1	1 0 1 0
Parent2	1 0 0 1	1 0 0 1	1 0 1 1
Fils 1	1 1 0 1	1 0 0 1	1 0 1 0
Fils 2	1 0 0 1	0 0 0 1	1 0 1 1

Fig. III. 8 Croisement en 2-points de deux chromosomes.

♦ *Croisement uniforme :*

Parent1 :	0	1	1	0	1	1	0	1
Parent2 :	1	0	0	0	1	0	1	1
Masque	0	1	0	1	0	0	1	1
Fils1 :	1	1	0	0	1	0	0	1
Fils2 :	0	0	1	0	1	1	1	1

Fig. III. 9 Croisement uniforme

III.9.6 Transfert

La mutation est définie comme la modification aléatoire d'une partie d'un chromosome. Il constitue une exploration aléatoire de l'espace des cordes [III.28].

C'est un phénomène qui a théoriquement un rôle plus marginal : il est là pour éviter une perte irréparable de diversité [III.29]. Différentes voies de mutation d'un chromosome sont également définies dans la littérature.

III.9.7 Valeurs des paramètres

Les paramètres qui conditionnent la convergence d'un algorithme génétique sont :

- la taille de la population d'individus ;
- le nombre maximum de générations ;
- la probabilité de croisement ;
- la probabilité de mutation.

Les valeurs de tels paramètres dépendent fortement du problème étudié. Il n'y a donc pas de paramètres adaptés pour résoudre tous les problèmes que l'on peut poser à un algorithme génétique. Cependant, certaines valeurs sont souvent utilisées (définies dans la littérature) et peuvent être de bons points de départ pour démarrer une recherche de solutions à l'aide d'un GA.

- la probabilité de croisement est choisie dans l'intervalle $[0,7, 0,99]$;
- la probabilité de mutation est choisie dans l'intervalle $[0,001,0,01]$.

Trouver de bonnes valeurs pour ces paramètres est donc un problème parfois délicat.

III.9.8 Demandes

Ayant été reconnu comme une approche valable pour les problèmes nécessitant une exploration efficace et économique sur le plan informatique, les algorithmes génétiques sont maintenant appliqués plus largement, aux entreprises, à la recherche scientifique en général, ainsi qu'à l'industrie. Les raisons de ce nombre croissant de demandes sont claires.

Ces algorithmes sont simples d'un point de vue informatique, mais très efficaces dans leur recherche d'amélioration [III.28].

III.10 Conclusion

Dans ce chapitre, nous avons commencé par une introduction aux algorithmes évolutionnaires, et nous avons vu que dans ces types d'algorithmes, il n'y a pas de leader qui donne des ordres à d'autres individus, ni de règles dictant un comportement global, mais l'évolution des individus est totalement aléatoire.

Ensuite, nous avons détaillé dans la deuxième partie une de ces grandes catégories, ce sont les algorithmes génétiques, qui ont l'énorme avantage de pouvoir être appliqués dans un grand nombre de domaines de recherche de solutions, quand ce n'est pas nécessaire pour avoir la solution optimale, qui prendrait par exemple trop de temps et de ressources à calculer (ou tout simplement si personne n'est capable de la trouver de manière théorique), il offre aussi aux programmeurs la liberté de choisir la comment mettre en œuvre différents procédés et la liberté de choisir les paramètres à utiliser puis de les modifier si les solutions obtenues ne sont pas bonnes.

Nous avons également noté certaines limites attachées à l'utilisation de ces algorithmes, comme le temps de calcul, le choix des paramètres, l'incertitude d'avoir atteint la solution optimale, etc....

CHAPITRE IV
SEPARATION AVEUGLE
DE SOURCES IMAGE
BASEE SUR
L'ALGORITHME HEP SO

IV.1 Introduction

Dans ce chapitre, un PSO modifié appelé HEP SO (High Exploration Particle Swarm Optimization) [IV.1] a été utilisé. Cependant, HEP SO s'appuie sur deux autres opérateurs. Le premier est basé sur le "multi-crossover" largement utilisé dans les algorithmes génétiques. Le second dépend du mécanisme de la colonie d'abeilles, mais les deux opérateurs ont été utilisés pour mettre à jour la vitesse et la position des particules, respectivement. Par conséquent, ces opérateurs sont utilisés pour trouver la matrice de séparation optimale.

La méthode proposée améliore l'efficacité globale des PSO standard en termes de bonne recherche et de bonnes performances. Les résultats expérimentaux confirment que la méthode SAS proposée démontre la robustesse et la précision de nombreux tests de signal d'image proposés sur la plate-forme "BSS demo".

IV.2 Le modèle du processus SAS

SAS est le processus utilisé pour séparer et récupérer le signal source du signal observé. Ce processus est géré par les propriétés statistiques du signal source. Le modèle SAS de base est illustré [IV.2].

L'algorithme sélectionné (HEP SO) optimise la fonction objectif pour rendre le signal de sortie indépendant.

Le modèle SAS est défini comme suit :

$$X = A.S \quad (IV.1)$$

Où $X = [x_1, x_2, x_3, \dots, x_m]^T$ est un vecteur contenant les signaux observés, $S = [s_1, s_2, s_3, \dots, s_n]^T$ est un vecteur contenant la source d'origine et A est une matrice de fusion complète classée par colonne. Le modèle linéaire peut être défini comme suit :

$$X_i(t) = a_{k1}s_1 + a_{k2}s_2 + \dots + a_{kn}s_n, \quad k = 1, n \quad (IV.2)$$

Supposons que le nombre n de sources est égal au nombre de mélanges m ($m = n$).

En fait, il existe deux hypothèses lors du dépannage des problèmes SAS. La première hypothèse est que les signaux sources sont statistiquement indépendants, la seconde hypothèse est Seule la source peut avoir une distribution gaussienne car la combinaison linéaire des signaux

gaussiens est gaussienne, auquel cas il est impossible de les séparer. Sous les hypothèses ci-dessus, les résultats SAS ont deux ambiguïtés inhérentes [IV.3].

1. Il n'est pas possible de déterminer la distribution des composants indépendants.
2. Il n'est pas possible de déterminer l'ordre des composants indépendants.

Dans cette étude, notre objectif principal est de séparer les signaux, l'ordre des sources estimés n'est pas important. La matrice de séparation (W_i) est donnée par l'équation (IV.3) :

$$y = W_i x \quad (IV.3)$$

IV.3 Évaluation de la fonction objective

La fonction de fitness proposée est basée sur kurtosis et l'information mutuelle. Kurtosis joue un rôle important dans le SAS. Il est utilisé pour mesurer la nature non gaussienne d'un signal et trier des composantes indépendantes. L'information mutuelle mesure les dépendances entre les composants. Pour obtenir l'indépendance des composants, la quantité d'informations mutuelles doit être minimisée. L'information mutuelle est définie comme suit :

$$I(y_1, y_2, \dots, y_n) = \sum_{i=1}^n H(y_i) - H(y) \quad (IV.4)$$

Ici, $H(y_i) = -E \log p_{y_i}(y_i)$ et $H(y) = -E \log p(y)$, La source sonore estimée peut être calculée à l'aide de la formule suivante.

$$kurtosis(y) = \sum_{i=1}^n |E(y_i^4) - 3E^2(y_i^2)| \quad (IV.5)$$

La fonction de fitness peut être exprimée comme suit :

$$J(y) = I(y_1, y_2, \dots, y_n) + \frac{1}{\sum_{i=1}^n |E(y_i^4) - 3E^2(y_i^2)|} \quad (IV.6)$$

Lorsque $J(y)$ est maximisé, les signaux de source audio estimés sont indépendants les uns des autres.

IV.4.1 Méthode Proposée [IV.19]

Dans la partie suivante, nous vous présenterons le système SAS et l'utiliserons pour résoudre le problème de la séparation sources d'images. Fondamentalement, le système SAS a été mis en œuvre en quatre étapes principales.

- (1) Le processus de mélange des sources d'image d'entrée.
- (2) Ensuite, opération de séparation est assuré par méthode HEPSO.
- (3) Ensuite, sources estimée est calculé en utilisant l'équation (IV.3).
- (4) Et enfin la performance de l'évaluation du système BSS. Le schéma de la méthode présentée est clairement illustré à la Figure IV.1.

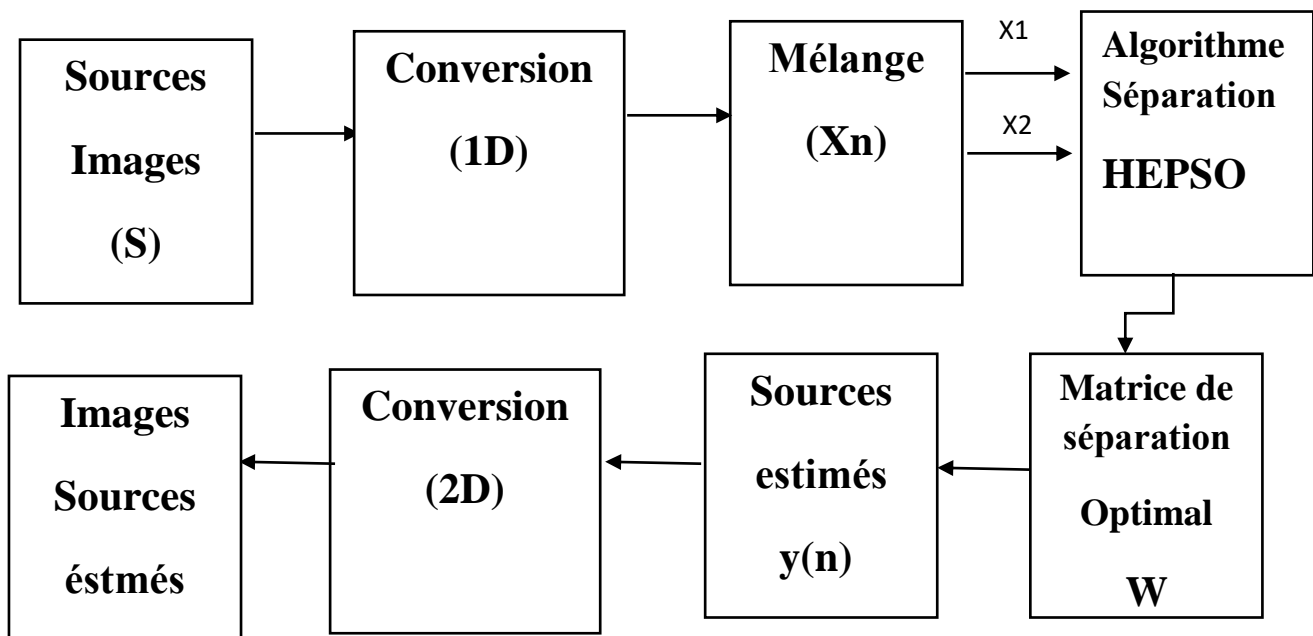


Fig. IV. 1 Schéma de la méthode proposée

IV.4.2 SAS est basé sur l'algorithme HEPSO

Dans cette partie, nous présenterons la méthode proposée pour SAS comme suit.

Etape 1 : la première population est générée.

Etape 2 : le poids inertiel (w) et le facteur d'apprentissage ($C1$, $C2$) sont calculés pour chaque itération.

Etape 3 : évaluez la fonction objective de toutes les particules.

Etape 4 : l'opérateur de colonie d'abeilles (Équation (IV.10)) n'est utilisé que si $p_1 < \text{écart-type}$ ou $p_2 < \frac{P_B \times t}{\text{maxiteration}}$, où p_1 et $p_2 \in [0,1]$, P_B sont la probabilité d'une colonie d'abeilles, t est l'itération actuelle et l'itération maximale est le nombre maximal d'itérations.

Etape 5 : générera un multi-cross over pour chaque particule qui n'a pas été sélectionnée lors de l'opération précédente. Pour les particules $v < P_c$, l'opérateur multi-crossover produit des vitesses ou de nouvelles particules via l'équation (IV.9). p_c est la probabilité de croisement multiple.

Etape 6 : les valeurs de x_{best} sont transformées pour former la matrice de séparation optimale(w)

Etape 7 : l'orthogonalisation est effectuée via l'équation (IV.7).

$$W_i = W_i \times \text{real}(\text{inv}(W_i \times W_i^T))^{-\frac{1}{2}} \quad (\text{IV.7})$$

Etape 8 : revient à l'étape 3 et répète le processus jusqu'à ce que les critères de terminaison soient remplis. Enfin, dans la neuvième étape, nous utiliserons l'équation (IV.3) pour calculer une autre source.

IV.4.3 Les Opérateurs

Dans cette partie présente deux opérateurs :

Le premier est basé sur trois chromosomes parents [IV.5]

Le second est le processus de la colonie d'abeilles [IV.6]. Ces opérateurs visent à améliorer le processus et à contourner les minimas locaux.

1.L'Opérateur 1

Cette étude propose une nouvelle méthode de croisement composée de trois chromosomes parents $[\theta_1, \theta_2, \theta_3]$, qui est différente de l'algorithme génétique. Un croisement multiple classique basé sur seulement deux chromosomes, proposé par Chang [IV.5]. On suppose que les chromosomes sont choisis au hasard dans la population. Si les chromosomes ont la qualité d'ajustement la plus faible, utilisez plusieurs croisements pour créer de nouveaux chromosomes.

$$\theta'_1 = \theta_1 + r(2\theta_1 - \theta_2 - \theta_3) \quad (\text{IV.8})$$

Où $r \in [0,1]$ est une valeur aléatoire. Le premier parent représente la meilleure position du groupe ($x_{gbest}(t)$), le deuxième parent représente la meilleure position de l'individu (x_{pbest}), et le troisième parent calcule la nouvelle vitesse de la particule sélectionnée. Utilisé. (IV.9)

$$v_i(t+1) = r \left[\frac{c2}{2} x_{gbest}(t) - x_{pbest}(t) - t \right] \tag{IV.9}$$

2. L'Opérateur 2

Il est basé sur l'algorithme optimisation Bee colony optimisation (BCO) [IV.6] [IV.7].

Dont la relation est donnée par l'équation (IV.10).

$$x_i^d(t+1) = x_i^d + (2r - 1) (x_i^d(t) - x_i^d(t)) \tag{IV.10}$$

Où $r \in [0,1]$ est une valeur aléatoire, d est un nombre aléatoire de longueur [1, dimension], et j est un nombre aléatoire de longueur [1, nombre de particules].

IV.4.4 Résultats Simulations

Pour illustrer les performances de la méthode proposée sur séparation images, des tests s sont effectués et comparés aux résultats de la méthode FastICA.

IV.4.4.1 Les signaux Source Images

Tableaux. IV. 1 Les signaux source de l'image test [IV.8]

Image de test	type du fichier	taille	ID de l'image					
			Dimensions	Largeur	Hauteur	Résolution horizontale	Résolution Vertical	Profondeur de Couleur
Lena	JPG(.jpg)	6.95Ko (7 119 Octetes)	225 × 225	225pixels	225pixels	96 ppp	96 ppp	24
Barbara	JPG(.jpg)	7.75Ko						

		(7 938 Octetes)	225 × 225	225pixels	225pixels	96 ppp	96 ppp	24
Einstein	JPG(.jpg)	16.4Ko (16.824 Octetes)	230 × 227	230pixels	227pixels	96 ppp	96 ppp	24
Ily	JPG(.jpg)	7.42Ko (7 605 Octetes)	259 × 194	259pixels	194pixels	96ppp	96ppp	24

La matrice de mélange 2x2 M est choisie aléatoirement comme suit :

$$M = \text{rand}(2,2) ;$$

IV.4.4.2 Les paramètres de simulation

Les valeurs d'initialisation des paramètres pour la méthode proposée par HEP SO sont les suivantes : La taille de la population est fixée à 100, le problème a une dimension de (dim= 4), la probabilité de mutation P_B est égale à 0,02, la probabilité de croisement P_c est fixée à 0,9 et les autres valeurs de combinaison sont fixés à $C_{1i} = 2.5$, $C_{2i} = 0.5$, $C_{1f} = 0.5$, $C_{2f} = 2.5$.

IV.4.4.3 Les mesures de performance

Pour d'examiner les performances de la technique proposée, nous l'évaluons à l'aide de trois indices de performance : valeur absolue des coefficients de corrélation (AVCC) [IV.9], PSNR [IV.10] et SSIM [IV.11]. Ils sont, respectivement définis comme suit :

a. Valeur absolue des coefficients de corrélation

$$AVCC = \left| \frac{\sum_{t=1}^M y_i(t) S_k(t)}{\sqrt{\sum_{t=1}^M y_i(t)^2 \sum_{t=1}^M S_k(t)^2}} \right|$$

b. Rapport signal/bruit de crête (Peak signal to noise ratio[PSNR]) :

Le PSNR est utilisé pour estimer la qualité de l'image et calculé entre la couverture et l'image filigranée, représenté par PSNR et l'erreur quadratique moyenne entre l'hôte et l'image filigranée est représentée par MSE [IV.10].

$$\text{PSNR} = 10 \log_{10}(d)^2 / \text{MSE}$$

Où d est la dynamique du signal. Dans le cas standard d'une image où les composantes d'un pixel sont codées sur 8 bits, $d = 255$.

$$\text{PSNR} = 10 \log_{10}(255)^2 / \text{MSE}$$

$$\text{MSE} = \frac{1}{N \times N} \sum_j^N \sum_j^N (A - A_w)^2$$

où A et A_w sont l'hôte et l'image en filigrane respectivement et $N \times N$ est la taille de l'image

$$\text{PSNR} = 10 \log \left[\frac{MN_{\max} s_{i,m,n} (m,n)^2}{\sum_{m=1}^M \sum_{n=1}^N s_i (m,n) - y_i (m,n)^2} \right]$$

Où $s_i(m, n)$ et $y_i(m, n)$, respectivement, désignent les valeurs de gris des images s_i et y_i dans (m, n) , et M et N sont les tailles des images. De plus, le nombre d'itérations peut également être utilisé pour mesurer les performances de l'algorithme.

Un PNSR supérieur représente les résultats de séparation préférés.

c La mesure de l'indice de similarité structurelle

(SSIM) est une méthode permettant de prédire la qualité perçue de la télévision numérique et des images cinématographiques, ainsi que d'autres types d'images et de vidéos numériques. SSIM est utilisé pour mesurer la similarité entre deux images. L'indice SSIM est une métrique de référence complète ; en d'autres termes, la mesure ou la prédiction de la qualité d'image est basée sur une image initiale non compressée ou sans distorsion comme référence.

L'indice SSIM est un développement des méthodes traditionnelles telles que le PSNR (rapport signal sur bruit maximal) et la méthode MSE, qui se sont révélées incompatibles avec la physiologie de la perception humaine. La différence avec d'autres techniques telles que MSE ou PSNR est que ces approches estiment les erreurs absolues. L'information structurelle est l'idée que les pixels ont de fortes interdépendances, en particulier lorsqu'ils sont spatialement proches. Ces dépendances contiennent des informations importantes sur la structure des objets dans la scène visuelle.

L'indice SSIM est calculé sur différentes fenêtres d'une image. La mesure entre deux fenêtres x et y de taille commune $N \times N$ est :

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Avec

μ_x la moyenne de x ;

μ_y la moyenne de y ;

σ_x^2 la varianc x ;

σ_y^2 la varianc y ;

σ_{xy} la covariance de x et y ;

$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ deux variables pour stabiliser la division avec un dénominateur faible ;

L la plage dynamique des valeurs de pixel (généralement $2^{\text{\#bits per pixel}} - 1$) ;

$k_1 = 0.01$ et $k_2 = 0.03$ par défaut

IV.5 Méthode Fast ICA

Simulation 1 : Image enstein et image ily

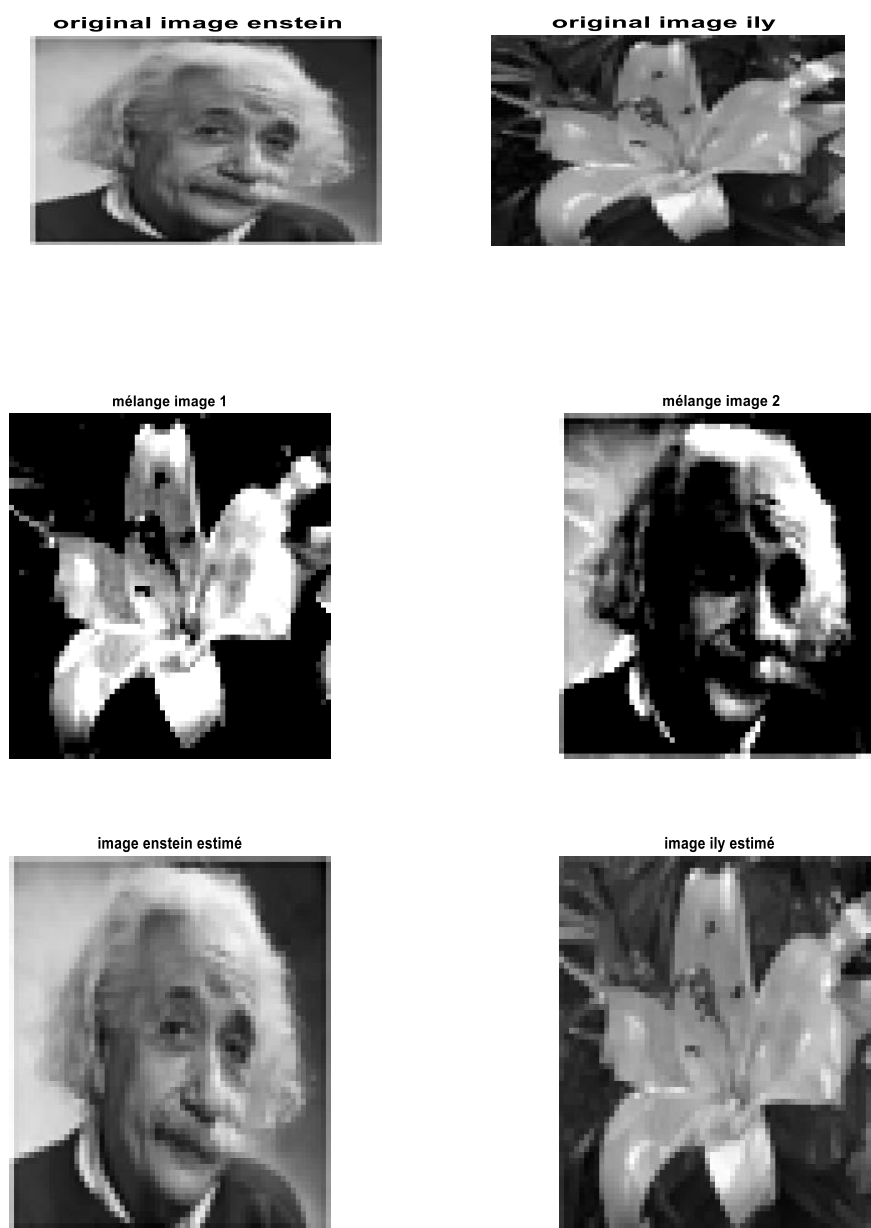
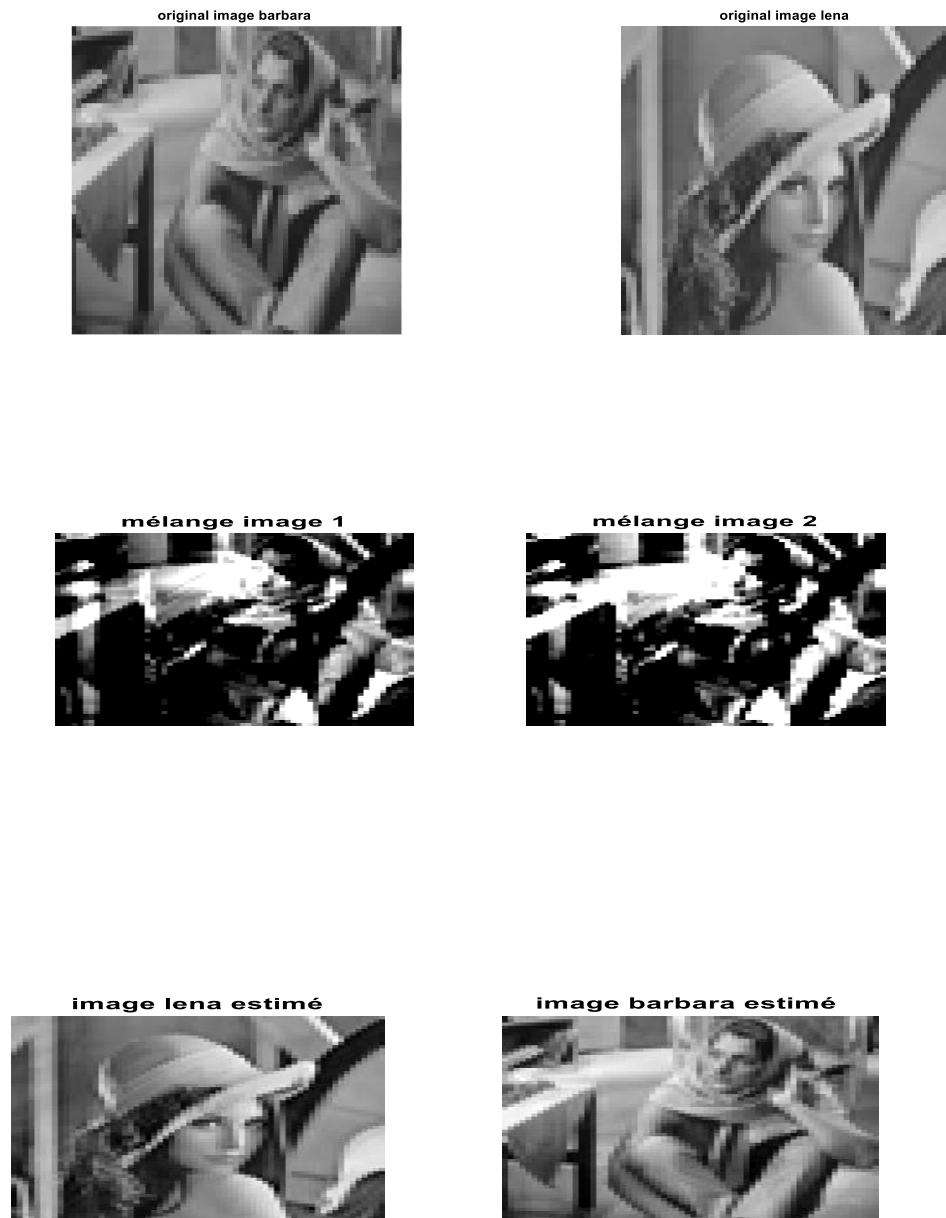


Fig. IV. 2 Images originales et éstms Enstein et ily

Simulation 2 : Image lena et image barbara**Fig. IV. 3 Images originales et éstmés Barbara et Lena**

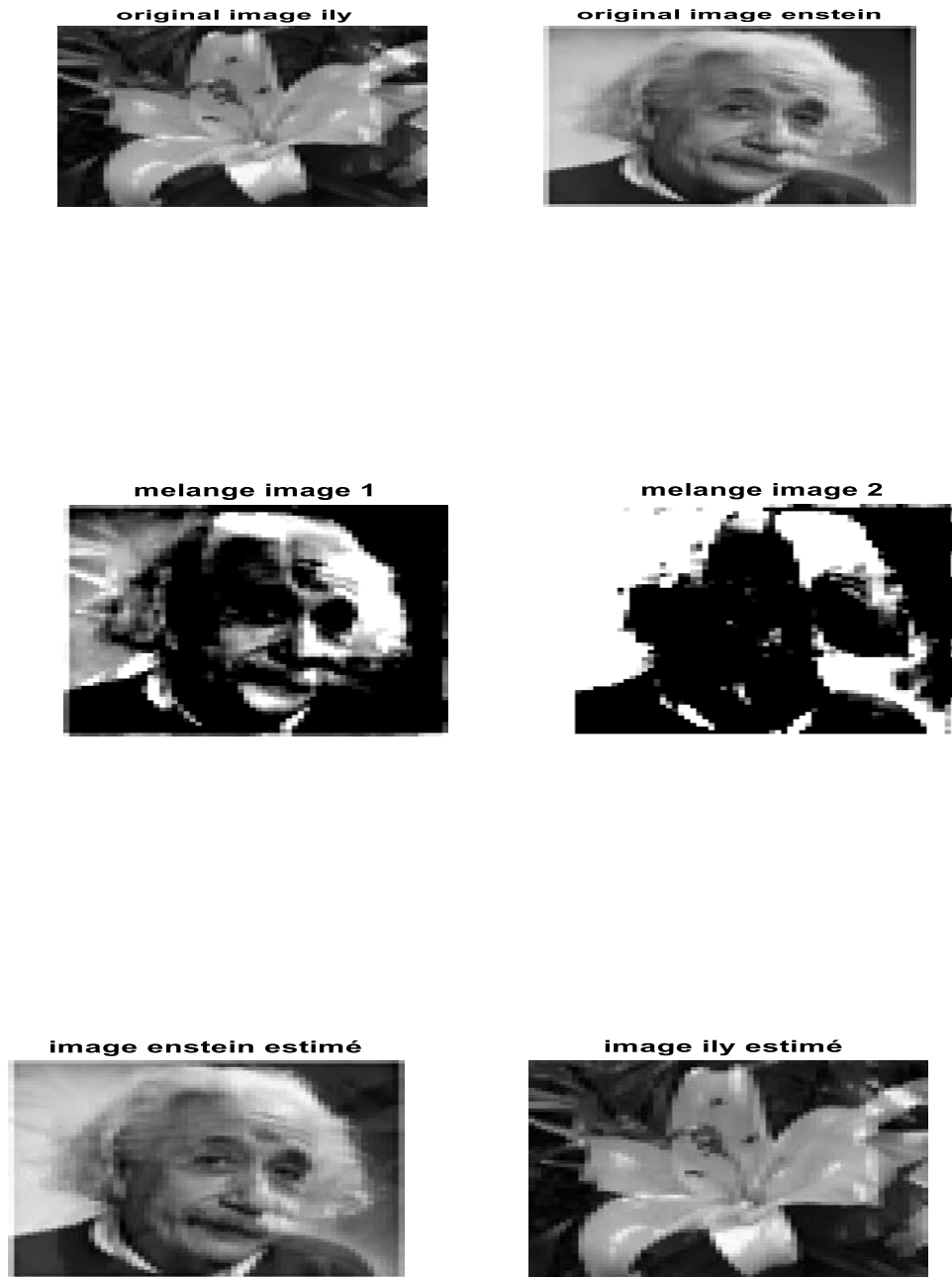
IV.6 Méthode HEP SO**Simulation 3 :Image enstein et image ily**

Fig. IV. 4 Images originales et éstms Einstein et ily

Simulation 4 :Image lena et image barbara**Fig. IV. 3 Images originales et éstmés Barbara et Lena**

IV.7 Interprétations

D'après les résultats expérimentaux, nous pouvons conclure ce qui suit :

Premièrement : les résultats du chapitre concernant les images estimées (récupérées) semblent presque similaires aux images originales

Deuxièmement: A travers les résultats de performance des algorithmes représentés dans les valeurs de PSNR, ssim, AVCC, qui sont résumées dans les tableaux présentés dans , il s'avère que l'algorithme HEPSO est optimal en termes de précision par rapport a la méthode FastICA.

Tableaux. IV. 2 PSNR (dB) pour l'algorithme Fast_ICA et HEPSO

Simulations	FASTICA	HEPSO
Simulation 1	Lena _ Barbara	
PSNR1	24.3193	26.1371
PSNR2	23.7849	24.0949
Simulation 2	ily _ Einstein	
PSNR1	30.0480	43.8829
PSNR2	24.9766	27.7895

Tableaux. IV. 3 ssim (dB) pour l'algorithme Fast_ICA et HEP SO

Simulations	FASTICA	HEPSO
Simulation 1	Lena _ Barbara	
ssim1	0.9493	0.9663
Ssim2	0.9568	0.9585
Simulation 2	ily _ Einstein	
ssim1	0.9927	0.9974
Ssim2	0.9093	0.9583

Tableaux. IV. 4 AVCC pour le HEPSO et l'algorithme Fast_ICA

Simulations 1	Lena _ Barbara			
Simulation	FASTICA		HEPSO	
	Y1	Y2	Y1	Y2
	0.9974	0.07118	0.9996	0.0441
	-0.0757	0.9971	0.0228	0.9988

Tableaux. IV. 5 AVCC pour le HEPSO et l'algorithme Fast_ICA

Simulations 2	ily _ Einstein			
Simulation	FASTICA		HEPSO	
	Y1	Y2	Y1	Y2
	0.9991	0.0426	-0.2741	0.9998
	-0.1644	0.9864	0.9880	-0.1000

D'après la comparaison des deux méthodes, nous avons clairement constaté que la méthode HEP SO surpasse largement la méthode classique représentée dans l'utilisation de FASTICA en termes de résultats des valeurs de PSNR (Tableaux. IV. 2), (Tableaux. IV. 3) ssim , ainsi qu'en termes de AVCC (Tableaux. IV. 4.) et (Tableaux. IV. 5)

Enfin, on peut dire que la méthode moderne HEP SO présentée et étudiée est plus précise et fiable

IV.5 Conclusion :

Dans ce dernier chapitre, nous avons présenté une technique d'optimisation utilisant HEP SO pour séparation source images. Pour améliorer les performances de séparation par rapport aux méthodes classiques utilisant gradient décent son inconvénient est la recherche s'arrête au premier minimum local rencontré. Les résultats de la simulation montrent une amélioration significative des performances Comparer avec l'algorithme FACTICA. Les perspectives d'avenir de ces travaux on pourra utiliser d'autres algorithmes optimisation évolutionnaires telle que PSO multiobjectis

Conclusion générale

Dans cette lettre, nous considérerons la séparation aveugle des sources d'images. Chacun nous raconte d'abord l'histoire du SAS et étudie la théorie actuelle. Comme toutes les simulations sont effectuées dans ce dernier, il révèle la source caractéristique d'un type particulier de mélange et visualise les propriétés du signal image. Cela était nécessaire pour tirer parti des méthodes SAS recherchées et développées pour cette tâche.

Dans le chapitre 2, nous avons introduit les concepts de base qui sont à la base de la compréhension des méthodes de traitement d'image, des différents types d'images, et brièvement introduit les concepts liés au domaine du traitement d'image utilisé en séparation aveugle à la source. Puis, au chapitre 3 on a montré la théorie description détaillée des types d'algorithmes évolutifs qui peuvent améliorer la technologie SAS. Dans le dernier chapitre, nous avons étudié la technique présentée par HEPSON pour améliorer les performances de séparation par rapport aux méthodes classiques utilisant gradient descent son inconvénient est la recherche s'arrête au premier minimum local rencontré. Les résultats de la simulation montrent une amélioration significative des performances Comparer avec l'algorithme FACTICA. Les perspectives d'avenir de ces travaux on pourra utiliser d'autres algorithmes optimisation évolutionnaires telle que PSO multiobjectifs.

Référence bibliographique

- [I.1] Ch. Jutten, J. Herault, A neuromimetic solution for the problem of sources discrimination, *Traitement du Signal. Grenoble série 5 Volume no 6* - 1988
- [I.2] A . Taleb and C. Jutten. «On underdetermined source separation». In *Proceedings ICASSP 99*, volume 3, pages 1445–1448, Arizona, Phoenix, March 1999.
- [I.3] A. Hyvannenn “ Survey on independent componenet Analysis”, *Neural computing surveye*, vol 2,pp 94-128, 1999
- [I.4] S. Amari, A. Cichocki”Adaptative Blind Signal Provesing- Neural Network ”. *proceedings of the IEEE*, Vol 86 N° 10, October 1998
- [I.5] A. Taleb and C. Jutten. «Source separation in post non linear mixture». *IEEE Transactions on signal Processing*, 47(10):2807–2820, October 1999.
- [I.6] Amari, S., Douglas, S.C., Cichocki, A. *et al.* (1997) Multichannel blind deconvolution and equalization using the natural gradient. *The 1st Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, Paris, France, pp. 101–104.
- [I.7] A. Taleb and C. Jutten: Nonlinear source separation: The post-nonlinear mixtures. *ESANN'1997*, pages 279-284, 1997.
- [I.8] A. Taleb and C. Jutten: Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, pages 2807-2820, 1999.
- [I.9] Y. Deville and S. Hosseini. Blind identification and separation methods for linear quadratic mixtures and/or linearly independent non-stationary signals. In *Proceedings of International Symposium on Signal Processing and its Applications (ISSPA)*, 2007.

- [I.10] S. Hosseini and Y. Deville. Blind separation of linear-quadratic mixtures of real sources using a recurrent structure. In Proceedings of International Work-Conference on Artificial and Natural Neural Networks, IWANN, pages 241-- 248, 2003.
- [I.11] S. Hosseini and Y. Deville. Blind maximum likelihood separation of a linear-quadratic mixture. In Proceedings of the International Workshop on Independent Component Analysis and Blind Source Separation (ICA), pages 694 --701, 2004.
- [I.12] M. Krob and M. Benidir. Blind identification of a linear-quadratic model using higherorder statistics. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 440 -- 443, 1993.
- [I.13] I. Meganem. Méthodes de Séparation Aveugle de Sources pour l'imagerie hyperspectrale. Application à la télédétection urbaine et à l'astrophysique. PhD thesis, Université Paul Sabatier Toulouse 3, 2012.
- [I.14] A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. IEEE Transactions on Signal Processing, pages 2807 _ 2820, 1999.
- [I.15] S. Amari, A. Cichocki "Adaptative Blind Signal Provesing- Neural Network ". proceedings of the IEEE, Vol 86 N° 10, October 1998
- [I.16] P. Comon and C. Jutten Eds. Handbook of Blind Source Separation : Independent Component Analysis and Applications. Elsevier, 2010.
- [I.17] Y. Deville. Blind source separation and blind mixture identification methods, In Wiley Encyclopedia of Electrical and Electronics Engineering. Wiley, 2016.
- [I.18] A. Hyvärinen, J. Karhunen, and E. Oja. Independent component analysis. Wileyinterscience,2001.
- [I.19] S. Theodoridis and K. Koutroumbas. Pattern Recognition. Academic Press, 2009.

- [I.20] P. Comon. Independent component analysis, a new concept ? Signal Processing, page287 _ 314, 1994.
- [I.21] C. Jutten and J. Herault. Blind separation of sources, part I : An adaptive algorithm based on neuromimetic architecture. Signal Processing, 24(1) :1 – 10, 1991.
- [I.22] A. Mansour and C. Jutten. What should we say about the kurtosis ? IEEE Signal Processing Letters, 6 :321–322, 1999.
- [I.23] A. Hyvärinen. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. IEEE Transactions on Neural Networks, 10(3) :626–634, 1999.
- [I.24] A. Hyvarinen , E. Oja, "A fast fixed-point algorithm for independent component analysis", Neural Comput 1997, 1483-92.
- [26] E. Moreau, O. Macchi “New self-adaptive algorithms for source separation based on contrast functions”, HOS93, IEEE Signal Processing Workshop on Higher Order Statistics, Lake Tahoe, USA, 215-219, 1993.
- [I.25] A. Belouchrani and J.-F. Cardoso. Maximum likelihood source separation for discrete sources. In Proceedings of EUSIPCO, volume 2, pages 768–771, 1994.
- [I.26] D.-T. Pham and P. Garat. Blind separation of mixtures of independent sources through a quasi maximum likelihood approach. IEEE Transactions on Signal Processing, 45(7) :1712–1725, 1997.
- [I.27] P. Comon. Independent component analysis, a new concept ? Signal Processing, 36(3) :287 – 314, 1994.
- [I.28] D.-T. Pham. Blind separation of instantaneous mixture of sources via the gaussian mutual information criterion. In Proceedings of EUSIPCO, pages 3–6, Tampere, Finland, 2000.
- [I.29] M. Babaie-Zadeh and C. Jutten. A general approach for mutual information minimization and its application to blind source separation. Signal Processing, 85(5) :975–995, 2005.

- [I.30] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non gaussian signals. IEE Proceedings-F Radar and Signal Processing, 140(6) :362–370, 1993.
- [I.31] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755) :788–791, 1999.
- [I.32] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755) :788–791, 1999.
- [I.33] P. Comon. Tensors, a brief introduction. IEEE Signal Processing Magazine, 31(3) :44–53, May 2014.
- [I.34] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. J. Math. and Phys., 6 :165–189, 1927.
- [I.35] R. A. Harshman. Foundation of the Parafac procedure : models and conditions for an explanatory multimodal factor analysis. UCLA Working papers in phonetics, 16 :1–84, 1970.
- [I.36] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari. Non negative matrix and tensor factorizations : Application to exploratory multi-way data analysis and blind separation. Wiley, 2009.
- [I.37] N. Murata, S. Ikeda, and A. Ziehe. An approach to blind source separation based on temporal structure of speech signals. Technical report 98-2, Riken Brain Science Institute, April 1998.
- [I.38] L. Parra and C. Spence. Convolutional blind source separation of non-stationary sources. IEEE Transactions on Speech and Audio Processing, 8(3) :320–327, May 2000.
- [I.39] A. Westner and V.-M. Bove. Applying blind source separation and deconvolution to realworld acoustic environments. In Proc. 106th of Audio Engineering Society, Munich, Germany,

1999.

[I.40] N. Mitianoudis and M. Davies. Audio source separation of convolutive mixtures. *IEEE Transactions on Speech and Audio Processing*, 11(5) :489–497, September 2003.

[I.41] W. Bobillet, E. Grivel, R. Guidorzi, and M. Najim. Cancelling convolutive and additive noises for speech enhancement. In *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, volume 2, pages 407–410, September 2003.

[I.42] J. M. F. Xavier, Victor A. N. Barroso, and J. M. F. Moura. Closed-form blind channel identification and source separation in sdma systems through correlative coding. *IEEE Journal on Selected Areas in Communications*, 16 :1506–1517, 1998.

[I.43] N. Charkani. Séparation auto-adaptative de sources pour les mélanges convolutifs. Application à la téléphonie mains-libres dans les voitures. PhD thesis, INP Grenoble, Novembre 1996.

[I.45] Dirk Callaerts. Signal Separation Methods based on Singular Value Decomposition and their Application to the Real-Time Extraction of the Fetal Electrocardiogram from Cutaneous Recordings. PhD thesis, K.U.Leuven - E.E. Dept., Dec. 1989.

[I.46] L. De Lathauwer, Callaerts D., B. De Moor, and J. Vandewalle. Fetal electrocardiogram extraction by source subspace separation. In *Proc. IEEE SP / Athos workshop*, pages 134–138, Girona, Spain, June 1995.

[I.47] X. Zeng, S. Li, G. Li, Y. Zhou, and D. Mo. Fetal ECG extraction by combining singlechannel svd and cyclostationarity-based blind source separation. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 6(4), August 2013.

[I.48] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation, Independent Component*

Analysis and Applications. Academic Press, Oxford UK, Burlington USA, 2010.

[I.49] V. Capdevielle, Ch. Servière, and J.-L. Lacoume. Blind separation of wide-band sources : application to rotating machine signals. In Proceeding of EUSIPCO 3, pages 2085–2088, 1996.

[I.50] P. Fabry. Techniques de séparation de sources appliquées à des signaux vibratoires de machines tournantes. PhD thesis, Grenoble, INPG, Grenoble, 2001.

[I.51] G. GELLE, M. COLAS, and C. SERVIERE. blind source separation : a tool for rotating machine monitoring by vibrations analysis. Journal of Sound and Vibration, 248(5) :865–885, 2001.

[I.52] N. Thirion, J. Mars, and C. Lacoume. Séparation aveugle de signaux large bande : Un nouveau challenge en prospection sismique. In 15ème Colloque du GRETSI, Groupe d’Etudes du Traitement du Signal et des Images, pages 1335–1338, Juan-les-Pins, France, septembre 1995.

[I.53] E. Chaumette, P. Comon, and D. Muller. Application of ica to airport surveillance. In Higher-Order Statistics, 1993., IEEE Signal Processing Workshop on, pages 210–214, 1993.

[I.54] M. Tria, M. Benidir, and E. Chaumette. blind separation of secondary radar signals using time-frequency analysis. In 16th European Signal Processing Conference (EUSIPCO 2008), pages 25–29, Lausanne, Switzerland, August 2008.

[I.55] L. Tomazeli Duarte, S. Moussaoui, and C. Jutten. Source separation in chemical analysis : Recent achievements and perspectives. IEEE Signal Processing Magazine, 31(3) :135–146, May 2014.

[I.56] Vincent, E. Musical source separation using time-frequency source priors. IEEE Transactions on Speech and Audio Processing, 14(1), 2005, 91–98.

- [I.57] Tukr, M., and Pentland, A. Face recognition using eigen faces. Proceedings of IEEE Conferences on Computer Vision and Pattern Recognition, Maui, HI, 1991, pp. 586–591.
- [I.58] Bartlett, M.S., Movellan, J.R., and Sejnowski, T.J. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6), 2002, 1450–1464.
- [I.59] Ma, J.C., Niu, Y.L., and Chen, H.Y. *Blind Signal Processing*, National Defense Industry Press, 2006, Beijing .
- [II.1] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson Prentice Hall, 2008 .
- [II.2] Anil K. Jain, Fundamentals of Digital Image Processing, 198
- [II.3] Mayet, F., Pinoli J.C. and Jourlin, M. Traitement du signal 13, 251-261, 1996
- [II.4] M Sandeli, traitement d'images par des approches bio-inspirées application à la segmentation d'images, université constantine 2. 2014.
- [II.5] CHAKIB, Reda. Acquisition et rendu 3D réaliste à partir de périphériques" grand public". 2018. PhD Thesis. Limoges.
- [II.6] C. Houassine, segmentation d'images par une approche biomimétique hybride. université université m'hamed bougara- boumerdes. 2012.
- [II.7] Image numérique. Numeriksciences. [Http://numeriksciences.fr](http://numeriksciences.fr),
- [II.8] K. Aounallah, les approches de segmentation d'image par coopération régions contours, 2010.
- [II.9] M .T. Chikh , amélioration des images par un modèle de réseau de neurones (comparaison avec les filtres de base), université abou-bakr belkaid tlemcen. 2011
- [II.10] Les images vectorielles et matricielles. <http://www.imedias.pro/cours-en-ligne/graphismedesign/definition-resolution-taille-image/les-images-vectorielles-matricielles/>.
- [II.11] O. Poutarédy. Différences entre image Bitmap et image vectorielle. Site des enseignants en Arts Appliqués de l'académie d'Orléans-Tours, 2015
- [II.12] Image file formats. Wikipedia, [Https://en.wikipedia.org/wiki/Image_file_formats](https://en.wikipedia.org/wiki/Image_file_formats)
- [II.13] R. Isdant. Traitement numérique de l'image. 2009

- [II.14] K. Aounallah, les approches de segmentation d'image par coopération régions contours, 2010.
- [III.1] Fogel, D. B. (1995), Evolutionary Computation: Toward a New Philosophy of Machine Intelligence (Piscataway, NJ: IEEE).
- [III.2] Liu M., & Hu H. (2013), Optimization of Image Zernike Moments Shape Feature Based on Evolutionary Computation, Intelligent Image and Video Interpretation: Algorithms and Applications, pp. 199-216.
- [III.3] Vergidis, K. (2008). Business Process Optimization Using An Evolutionary Multi Objective Framework: A thesis in computer science (phd thesis). Cranfield university, School of applied sciences, department of manufacturing, UK.
- [III.4] Skrlec, D. (2010), Smart Long-Term Planning For Smartgrids. In Energy Conference and Exhibition (EnergyCon), 2010 IEEE International (pp. 682- 687). IEEE.
- [III.5] Tiwari, A., & Roy, R. (2004). Challenges In Real World Optimisation Using Evolutionary Computing.
- [III.6] Coello Coello, C.A. (2000), Handling Preferences in Evolutionary Multi-objective Optimization: A Survey, in Proceedings of the Congress on Evolutionary Computation (CEC'2000) San Diego, USA, pp. 30-37.
- [III.7] M. Hamchaoui, "Conception économique des assemblages en Construction Métallique". Thèse de doctorat, Université de Savoie, pp.4-60,1997.
- [III.8] Jayachandran, "Design of tall buildings: Preliminary Design and Optimisation", International Conference on Tall Buildings and Industrial Structures, Coimbatore, India, Keynote Lecture, pp.54 ,2003.
- [III.9] [63] É. Goubault. "Cours sur les Algorithmes Évolutionnaires et Problèmes Inverses", École Polytechnique ParisTech, France.
- [III.10] Marc Schoenauer (<http://eodev.sourceforge.net/>)
- [III.11] GLOVER, Fred. Future paths for integer programming and links to artificial intelligence. Computers & operations research, 1986, vol. 13, no 5, p. 533-549.

- [III.12] Zbigniew Michalewicz, David B. Fogel, *how to Solve It: Modern Heuristics*, Second Edition, Chapter 1, 2, 5, 6, Springer, 2004.
- [III.13] Glover F. Tabu search—part I, *ORSA Journal on Computing*, 1(3) (1989) pp. 190–206
- [III.14] Glover F., Tabu search—part II, *ORSA Journal on Computing*, 2(1) (1990) pp. 4–32.
- [III.15] Tai-Hsi Wu, Jinn-Yi Yeh, Chin-Chih Chang, *A hybrid Tabu Search Algorithm to Cell Formation Problem and its Variants*, *World Academy of Science, Engineering and technology* 53 (2009) pp. 1090–1094.
- [III.16] D. Simon, “Biogeography-Based Optimization,” *IEEE Trans. Evol. Comput.*, vol. 12 no. 6, pp. 702–713, 2008.
- [III.17] R. MacArthur and E. Wilson, *The Theory of Biogeography*. Princeton University Press, Princeton, NJ, 1967.
- [III.18] R. Ricklefs, and G. Miller, *Ecologie*. éd. de Boeck, 2005. ISBN 2-7445-0145-X.
- [III.19] H. Ma, “An analysis of the equilibrium of migration models for biogeography-based optimization,” *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.
- [III.20] T. Bäck, *Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996. ISBN 978-0-19-509971-3.
- [III.21] Abraham, Siby, KISS, Imre, SANYAL, Sugata, et al. *Steepest Ascent Hill Climbing for A Mathematical Problem*. ArXiv preprint arXiv:1010.0298, 2010.
- [III.22] Skiena, Steven S. *Combinatorial search and heuristic methods*. In: *The Algorithm Design Manual*. Springer, London, 2012. p. 230-272.
- [III.23] Arts, Emile, Korst, Jan, et Michiels, Wil. *Simulated annealing*. In: *Search methodologies*. Springer, Boston, MA, 2005. p. 187-210.
- [III.24] METROPOLIS, Nicholas, ROSENBLUTH, Arianna W., ROSENBLUTH, Marshall N., et al. *Equation of state calculations by fast computing machines*. *The journal of chemical physics*, 1953, vol. 21, no 6, p. 1087-1092.

[III.25] Szu, Harold et Hartely, Ralph. Fast simulated annealing. Physics letters A, 1987, vol.122, no 3-4, p. 157-162.

[III.26] CURILEF, Sergio. Generalized statistical mechanics for the N-body quantum problem ideal gases. Zeitschrift für Physik B Condensed Matter, 1997, vol. 100, no 3, p. 433-440.

[III.27] Dalligna Junior, Alcino, Silva, Renato S., Mundim, Kleber C., et al. Performance and parameterization of the algorithm Simplified Generalized Simulated Annealing. Genetics and Molecular Biology, 2004, vol. 27, no 4, p. 616-622.

[III.28] David E.Goldberg. "Algorithmes génétiques Exploitation, optimisation et apprentissage automatique". Addison-Wesley, France, SA, 1994.

[III.29] Jean-Michel Renders. "Algorithmes génétiques et réseaux de neurones". Hermes, 1995.

[III.30] Blaise Madeline. "Algorithmes évolutionnaires et résolution de problèmes de satisfaction de contraintes en domaines finis". Thèse de

Doctorat en sciences, Université de Nice-Sophia antipolis, 2002.

[III.31] Youssef Harrat. "Contribution à l'ordonnancement conjoint de la production et de la maintenance : Application au cas d'un job Shop".

Thèse de Doctorat, L'U.F.R des Sciences et Techniques, Université de Franche-comté, 2003.

[III.32] Pascal Rebreyend. "Algorithmes génétiques hybrides en optimisation combinatoire". Thèse de Doctorat, Ecole Normale

Supérieure de Lyon, 1999.

[III.33] Claude Alazard et Sabine Separi. "Contrôle de gestion", Manuel et applications. D E C F. Paris , Dunod, 2001.

[III.34] Michèle Sebag et Marc Schoenauer. "Contrôle d'un algorithme génétique". Revue d'intelligence artificielle Vol.10/1996.

Chapitre IV

[IV.1] M.J. Mahmoodabadi, Z. SalahshoorMottaghi, A. Bagheri. "HEPSO: High exploration particle swarm Optimization", information science 2014, 101- 111

- [IV.2] R.Niva Das, Aurobinda Routray, Pradipta K. Dash. "ICA Methods for Blind Source Separation of Instantaneous Mixtures: A Case Study", Neural Information Processing–Letters and Reviews, November 2007 Vol. 11, No.11.
- [IV.3] Mazur and A. Mertins, "A sparsity-based the criterion for Solving the permutation ambiguity in convolutive blind source separation", in Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, Prague, Czech Republic, May 2011, p. 1996– 1999.
- [IV.4] Khalfa, Ali, et al. Blind Audio Source Separation Based On High Exploration Particle Swarm Optimization. KSII Transactions on Internet & Information Systems, 2019.
- [IV.5] W.D. Chang, "A multi-crossover genetic approach to multivariable PID controllers tuning", Expert Syst. Appl. 2007; 620–626.
- [IV.6] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", Inf. Sci. 2012, 120–142.
- [IV.7] Chien-Ching Chiu, Chien-Hung Chen & Yu-Sheng Fan, "Image Reconstruction of a Buried Conductor by Modified Particle Swarm Optimization", IETE Journal of Research, 2012, 58:4, 284-291.
- [IV.8] T.D. Seeley, "The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies", Harvard University Press,
- [IV.9] KHALFA, Ali, et al. Blind Audio Source Separation Based On High Exploration Particle Swarm Optimization. 2019
- [IV.10] . Ray-Shine Run, Shi-Jinn Horng , Jui-Lin Lai, Tzong-Wang Kao, Rong-Jian Chen, "An improved SVD-based watermarking technique for copyright protection," in Expert Systems with Applications 39 (2012) 673–689
- [IV.11] <https://vicuesoft.com/glossary/term/ssim-ms-ssim/>