

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

De Master Académique en Automatique
Option : Commande des systèmes

Thème

**Seuillage d'histogrammes basé sur un Algorithme à estimation
de distribution**

Proposé et dirigé par :

Mr HAMMOUCHE.K

Présenté par :

Mr MELLAL Lyès

Soutenu le : 04 /09 /2012

Promotion 2012

Ce travail a été préparé à : L'université Mouloud Mammeri de Tizi-Ouzou

Remerciements

Je remercie tous les membres de jury.

Je tiens à remercier Mr HAMMOUCHE Kamal en lui exprimant une sincère reconnaissance pour son aide et ses conseils tout au long de ce travail.

Mes remerciements à toute ma famille ainsi que tous mes amis pour leurs encouragements.

Dieu MERCI

MELLAL Lyès

DEDICACES

Je dédie ce travail à la mémoire de mon grand père, à mes parents, à mes amis et à toute la famille.

Sommaire

Introduction générale.....	1
Chapitre 1: Notions sur la segmentation	
1.1 Introduction.....	3
1.2 Définition de la segmentation.....	3
1.3 Les Différentes approches de segmentation.....	3
1.3.1 Approche frontière.....	4
1.3.2 Approche région.....	4
1.3.2.1 Agrégation de pixels (croissance de régions).....	4
1.3.2.2 Méthode par division (partage)	6
1.3.2.3 Méthodes par fusion	7
1.3.2.4 Méthodes par division-fusion (Split and merge).....	8
1.3.3 Approche par classification des pixels.....	9
1.4 Segmentation d'image par seuillage	9
1.4.1 Définition du seuillage.....	10
1.4.2 Seuillage local	10
1.4.3 Seuillage global.....	11
1.4.3.1 Méthode d'Otsu.....	12
1.4.3.2 Méthode de seuillage basée sur l'entropie.....	13
1.5 Problème du multi-seuillage	14
1.6 Conclusion	14
Chapitre 2: Les Algorithmes à estimation de distribution	
2.1 Introduction.....	15
2.2 Description	15
2.3 Les différentes étapes de L'AED.....	16
2.4 Rappels sur les distributions de probabilités jointes.....	17
2.4.1 Les différentes lois de probabilités	18
2.5 Les différents AEDs.....	22
2.5.1 Modèle monovarié.....	22
2.5.1.1 Algorithme de distribution marginale monovarié.....	22
2.5.1.2 La population basée sur l'apprentissage incrémental (PBIL)	22
2.5.1.3 Algorithme génétique compact (CGA)	23
2.5.2 Modèle bivarié.....	24
2.5.2.1 Maximisation de l'information mutuelle.....	24

2.5.2.2 Combinaison des Optimiseurs.....	25
2.5.2.3 Algorithme de distribution marginale bivariable	25
2.5.3 Modèle multivariable	26
2.5.3.1 Algorithme génétique compact étendu (ECGA).....	27
2.5.3.2 Algorithme de distribution factorisée (FDA)	27
2.5.3.3 Algorithme d'optimisation Bayésienne (BOA).....	28
2.6 Conclusion	30

Chapitre 3: Seuillage d'histogramme basé sur L'AED

3.1 Introduction.....	31
3.2 Seuillage d'histogramme à base de L'AED	31
3.3 Tests et résultats	32
3.3.1 Résultats expérimentaux.....	34
3.3.2 Interprétation des résultats	50
3.4 Conclusion	50
Conclusion générale	51

Liste des tableaux

- Tableau (2.1) : Comparaison des différentes variantes de L'AED.
- Tableau (3.1) : Paramètres et fonctions de densités de probabilités des 6 distributions marginales.
- Tableau (3.2) : Valeur des seuils obtenus par une recherche exhaustive.
- Tableau (3.3) : Résultats du seuillage à base de L'AED et selon la loi normale.
- Tableau (3.4) : Résultats du seuillage à base de L'AED et selon la loi gamma.
- Tableau (3.5) : Résultats du seuillage à base de L'AED et selon la loi log- normale.
- Tableau (3.6) : Résultats du seuillage à base de L'AED et selon la loi exponentielle.
- Tableau (3.7) : Résultats du seuillage à base de L'AED et selon la loi de Weibull.
- Tableau (3.8) : Résultats du seuillage à base de L'AED et selon la loi de Rayleigh.

Liste des figures

- Figure (1.1) : Division d'une région inhomogène R en quatre quadrants R_k $k=1,\dots,4$.
- Figure (1.2) : Exemple d'application de l'algorithme de division –fusion.
- Figure (1.3) : Description du découpage de l'image au moyen d'un arbre quaternaire.
- Figure (1.4) : Image réelle (a) et son histogramme (b).
- Figure (2.1) : Représentation graphique du modèle de probabilité en supposant la dépendance d'ordre deux entre les variables.
- Figure (2.2): Représentation graphique du modèle de probabilité multivariable.
- Figure (3.1) : Images Tests.
- Figure (3.2) : Histogrammes des images tests.
- Figure (3.3) : Images Lena segmentées en utilisant la Loi Normale.
- Figure (3.4) : Images Lac segmentées en utilisant la Loi Normale.
- Figure (3.5) : Images Lena segmentées en utilisant la Loi Gamma.
- Figure (3.6) : Images Lac segmentées en utilisant la Loi Gamma.
- Figure (3.7) : Images Lena segmentées en utilisant la Loi Log-Normale.
- Figure (3.8) : Images Lac segmentées en utilisant la Loi Log-Normale.
- Figure (3.9) : Images Lena segmentées en utilisant la Loi exponentielle.
- Figure (3.10) : Images Lac segmentées en utilisant la Loi exponentielle.
- Figure (3.11) : Images Lena segmentées en utilisant la Loi Weibull.
- Figure (3.12) : Images Lac segmentées en utilisant la Loi Weibull.
- Figure (3.13) : Images Lena segmentées en utilisant la Loi Rayleigh.
- Figure (3.14) : Images Lac segmentées en utilisant la Loi Rayleigh.

Introduction générale

Introduction générale

Le traitement d'image désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Le traitement d'images commence à être étudié dans les années 1920 pour la transmission des images par un câble sous-marin allant de New York à Londres et le véritable essor n'a lieu que dans les années 1960 grâce à l'avènement des ordinateurs. En 1980, David Marr formalise le premier les détections de contours, et dans les années 1990, la recherche médicale devient le plus gros demandeur en traitement d'images pour améliorer les diagnostics faits à partir des nombreuses techniques utilisées en imagerie médicale.

Pour traiter une image, il faut appliquer des opérateurs parmi lesquels, on distingue la segmentation.

La segmentation est l'un des opérateurs qui s'appliquent sur une image, elle permet d'identifier les composantes de l'image. Plusieurs approches ont été proposées, parmi elle la segmentation à base de seuillage d'un histogramme. Cette approche est la plus simple et la plus utilisée.

Le seuillage constitue une approche simple à la segmentation, elle est facile à mettre en œuvre et efficace dans les systèmes en temps réel. Les méthodes par seuillage consistent à affecter à chaque point d'image un numéro de classe, par comparaison des valeurs des niveaux de gris de ces points à des seuils déterminés préalablement en optimisant un critère donné.

Le problème du seuillage consiste à calculer la valeur d'un ou plusieurs seuils en optimisant un critère donné. La recherche d'un seul seuil peut se faire facilement et rapidement d'une manière exhaustive. Cependant cette recherche exhaustive devient prohibitive lorsque le nombre de seuils augmente.

Pour résoudre ce problème, plusieurs techniques ont été proposées. Parmi elles, on peut citer les méthodes métaheuristiques d'optimisation.

Dans le cadre de notre travail, on va s'intéresser à l'étude de seuillage d'un histogramme à base d'un algorithme à estimation de distribution. Cet algorithme est une nouvelle métaheuristique permettant de résoudre un problème d'optimisation pour obtenir une solution optimale qui dans notre cas représente les seuils d'un histogramme.

L'algorithme à estimation de distribution est une classe des algorithmes évolutionnaires, il est utilisé pour résoudre les problèmes d'optimisation. Il utilise les informations collectées tout au long du processus d'optimisation pour construire un modèle probabiliste afin de générer de nouvelles solutions. La nouvelle population est engendrée en utilisant une distribution de probabilité estimée à partir des solutions de la génération précédente.

Ce mémoire présente une étude comparative entre plusieurs techniques de seuillage à base de l'algorithme à estimation de distribution et en modifiant son modèle probabiliste.

Ce mémoire contient trois chapitres et sont organisés comme suit :

Le premier expose les notions de base sur la segmentation en générale et le seuillage en particulier.

Le deuxième chapitre est consacré à la description de l'algorithme à estimation de distribution.

L'application de l'algorithme à estimation de distribution pour résoudre le problème du seuillage est présentée dans le dernier chapitre. Les résultats des tests obtenus par l'algorithme à estimation de distribution selon différents modèles probabilistes monovariés seront également présentés.

Chapitre 1

1.1 Introduction

L'objectif de ce chapitre est d'introduire la définition, le but et les différentes approches de la segmentation et d'approfondir dans l'étude de la segmentation par seuillage.

La segmentation joue un rôle prépondérant dans le traitement d'image. Elle fait partie d'une chaîne de traitement du signal et elle est réalisée avant les étapes d'analyse et de prise de décision dans plusieurs processus d'analyse d'image, tel que la détection des objets. Elle sert à localiser et délimiter les entités présentes dans l'image.

1.2 Définition de la segmentation

Par définition, la segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image A en sous ensembles R_i , appelés régions, tels que :

$$\begin{aligned}\forall i; R_i &\neq \emptyset \\ \forall i, j; i \neq j & R_i \cap R_j = \emptyset \\ A &= \cup_i R_i\end{aligned}$$

Une région est un ensemble connexe de points image (pixels) ayant des propriétés communes (intensité, texture...) qui les différencient des pixels des régions voisines. Les connaissances utilisées sont le plus souvent du domaine de l'image numérique et du traitement du signal, donc sémantiquement assez pauvre.

Il n'y a pas de méthode unique de segmentation d'une image, le choix d'une technique est lié :

- A la nature de l'image (Eclairage non homogène, reflets, Présence de bruit, de zones texturées, Contours flous, en partie occultée).
- Aux opérations situées en aval de la segmentation (Localisation, mesure, calcul 3D, Reconnaissance de formes, interprétation, Diagnostic, contrôle de qualité).
- Aux primitives à extraire (Contours, segments de droite, angles, Régions, formes, Textures).
- Aux contraintes d'exploitation (Complexité algorithmique, fonctionnement en temps réel, Taille de la mémoire disponible en machine).

Du fait de cette diversité, il est difficile de définir, de manière absolue, une "bonne" segmentation. La segmentation n'est pas une fin en soi, sa qualité est fonction des résultats obtenus par les traitements situés en aval.

1.3 Les Différentes approches de segmentation

La segmentation fait référence aux notions de différence et de similarité comme les perçoit le système visuel humain et ceci donne naissance à trois approches couramment qualifiées d'approche "frontière", d'approche "région" et classification des pixels :

1.3.1 Approche frontière

La notion de frontière est associée à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles connexes de points.

Les techniques de détection de contours sont les plus utilisées dans l'approche frontière. Les détecteurs de contours utilisés peuvent être simples, comme les opérateurs de Sobel ou de Roberts, ou plus complexe tel que l'opérateur de Canny. Ces méthodes ne conduisent pas directement à une segmentation de l'image telle qu'on l'a définie précédemment, car les contours obtenus sont rarement connexes, il faut donc procéder à une fermeture des contours si l'on souhaite une partition complexe de l'image. En effet, après fermeture de contours, la dualité contours- régions apparaît nettement. Les régions sont définies topologiquement comme l'intérieur d'une ligne fermée.

1.3.2 Approche région

La notion de région fait référence à des groupements de points (pixels) ayant des propriétés communes. Les méthodes de l'approche région aboutissent directement à une partition de l'image où chaque pixel est affecté à une région unique. La croissance par régions et la division-fusion sont des techniques les plus courantes de l'approche région.

1.3.2.1 Agrégation de pixels (croissance de régions)

La méthode d'agrégation de pixels, également appelée croissance de régions, est une technique ascendante qui, partant de la représentation de l'image comme un ensemble de pixels, les regroupe selon un double critère d'homogénéité et d'adjacence. Cette croissance est conduite par l'utilisation d'un mode de contrôle défini a priori et appelé prédicat.

Le prédicat, noté $pred$ par la suite, permet d'identifier une contrainte que doivent satisfaire les régions, il peut être de nature géométrique (région rectangulaire, région convexe..), de nature radiométrique (région homogène..) ou combinaison de deux.

Comme exemples de prédicat d'homogénéité pour une région R , on peut citer :

-La variance $\sigma^2(R)$ des niveaux de gris de l'image associés aux pixels de la région R est inférieure à un seuil préfixé.

-La proportion $\alpha(R)$ de pixels dont le niveau de gris se situe hors de l'intervalle $[m(R) - \sigma(R), m(R) + \sigma(R)]$, où $m(R)$ est la valeur moyenne des niveaux de gris, ne dépasse pas un seuil préfixé.

Le prédicat est une fonction booléenne sur chaque région.

Le processus de création des régions considère les régions les unes après les autres et comprend, pour chaque région, une phase d'initialisation et une phase itérative.

Pour présenter les deux phases, on suppose que l'ensemble des pixels S de l'image a été partiellement partitionné en $i-1$ régions R_j , $j=1, \dots, i-1$. On s'intéresse à la génération de la région R_i qui sera construite dans l'ensemble $S^{(0)} = S - \bigcup_{j=1}^{i-1} R_j$.

Le choix d'un germe d'une région $R_i^{(0)}$ peut se faire par seuillage sur un attribut (par exemple le niveau de gris), ou de toute autre manière (extraction de formes géométrique, etc). Un germe peut éventuellement être réduit à un seul point.

La construction de $R_i^{(k+1)}$ s'effectue à partir de $R_i^{(k)}$, en lui adjoignant des pixels de $S^{(k+1)}$ défini par $S^{(k+1)} = S^{(k)} - R_i^{(k)}$, suivant la relation :

$$R_i^{(k+1)} = R_i^{(k)} \cup \left\{ s \in S^{(k+1)} \setminus S \mid \mathfrak{R}_1 R_i^{(k)} \text{ et } s \mathfrak{R}_2 R_i^{(k)} \right\}$$

Où \mathfrak{R}_1 est une relation de type similarité radiométrique entre un pixel et la région croissante $R_i^{(k)}$ et où \mathfrak{R}_2 est une relation de type connexité entre un pixel candidat et la région croissante.

➤ Remarque

La relation \mathfrak{R}_1 pour l'élaboration d'un ensemble X peut être spécifiée comme suit :

Calcul de caractéristiques images (texture, intensité, couleur) sur l'ensemble des pixels agrégés à l'itération précédente, c'est-à-dire l'ensemble $R_i^{(k)} \setminus R_i^{(k-1)}$. Pour la première étape, cet ensemble est pris égal à $R_i^{(0)}$.

Le fait d'utiliser la caractéristique moyenne sur l'ensemble des pixels retenus à l'itération précédente $\{R_i^{(k)} \setminus R_i^{(k-1)}\}$ et non sur l'ensemble des pixels de $R_i^{(k)}$ permet de satisfaire l'hypothèse d'homogénéité locale de la classe en question, tout en autorisant des variations possibles au niveau global de cette classe.

Pour chacun des points de $S^{(k+1)}$, on calcule la distance à la caractéristique moyenne prédéfinie. Un point s de $S^{(k+1)}$ est désigné comme point candidat à l'agrégation si la valeur de distance est inférieure à un seuil prédéfini.

L'algorithme donné dans la table (1.1) résume le processus de segmentation par croissance des régions.

```

i :=0
/*a-initialisation de la région Ri(0)*/
k :=0 ; /*le numéro de l'itération*/
si i=1 alors S(0)=S SINON S(0)= S-Uj=1i-1 Rj
choix d'un germe Ri(0)
/*b-phase itérative : construction des itérés Ri(k)*/
/*mise à jour de l'ensemble S(k) des points candidats à l'agrégation*/
S(k+1)=S(k)-Ri(k) ;
Construction de Ri(k+1) à partir de Ri(k) et S(k+1) ;
Si pred(Ri(k+1)) alors
    Si S(k+1)=∅
alors arrêt/* toutes les régions vérifient le prédicat*/
    SINON
        DEBUT
            Ri=Ri(k+1) ;
            i :=i+1 ;
            ALLER en a) ;
            /*initialisation d'une nouvelle région*/
        FIN
SINON
    Si S(k+1)=∅ ALORS
        ARRET/*i-1 régions vérifient le prédicat,
            La région i comprenant les points restants n'est pas homogène*/
    SINON
        k=k+1;
        ALLER en b) ;/*poursuite de la croissance*/

```

Table (1.1): Algorithme de segmentation par croissance de régions

1.3.2.2 Méthode par division (partage)

Ces méthodes s'attachent à diviser l'image en région d'une manière récursive. Initialement, une mesure d'inhomogénéité $E(R)$ est calculée sur l'unique région que forme l'image originale au moyen de l'équation :

$$E(R) = \frac{1}{N(R)} \sum_{i=1}^{N(R)} (x_i - v(R))^2$$

Où $N(R)$: représente le nombre de pixels de la région R .

x_i : représente le niveau de gris du pixel i .

$v(R)$: valeur moyenne des niveaux de gris des pixels de la région R donnée par :

$$v(R) = \frac{\sum_{i=1}^{N(R)} x_i}{N(R)}$$

Cette mesure d'inhomogénéité $E(R)$ représente le degré de variation du niveau de gris de chaque pixel de la région. Le but du prédicat d'uniformité $P(R)$ est de décider si la région R est homogène ou non.

Cette décision binaire s'obtient généralement en comparant $E(R)$ à un seuil T comme l'indique la relation suivante :

$$P(R) = \begin{cases} \text{vrai} & \text{si } E(R) < T \\ \text{faux} & \text{sinon} \end{cases}$$

Dans le cas homogène, la région R est laissée telle quelle, alors que dans le cas inhomogène, la région R est divisée en sous régions plus homogènes ; cette opération peut s'effectuer, par exemple en divisant la région R en quatre quadrants d'égales surfaces R_i , $i=1, \dots, 4$ comme l'illustre la figure (1.1),

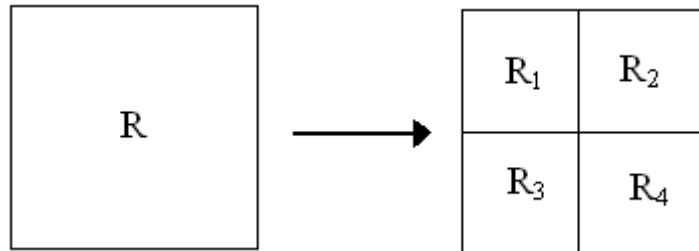


Figure (1.1) : Division d'une région inhomogène R en quatre quadrants R_k $k=1, \dots,$

Le processus de calcul de la mesure d'inhomogénéité $E(R)$, de l'évaluation du prédicat d'uniformité $P(R)$ et de la division de quatre quadrants, s'effectue récursivement pour chacune des régions trouvées quelle que soit sa taille, jusqu'à ce que toutes les régions soient homogènes.

1.3.2.3 Méthodes par fusion

Cette méthode réunit les différentes régions adjacentes en une nouvelle région, si la mesure d'inhomogénéité de cette dernière ne dépasse pas un certain seuil T . L'arrêt de cette opération est atteint lorsqu'un certain critère de fin de rassemblement est satisfait. Par exemple lorsque la somme des erreurs quadratiques SEQ calculée entre l'image originale et l'image approchée, excède un certain seuil T . Ce critère est décrit formellement par :

$$SEQ(K) = \sum_{i=1}^K \sum_{x_j \in R_i} |x_j - v(R_i)|^2,$$

avec K le nombre de régions et $v(R_i)$ est le centre de la classe C_i ;

Cette méthode est considérée comme une opération de post traitement qui vient compléter un algorithme de segmentation. Elle peut être modélisée au moyen du graphe de contiguïtés des régions GCR (Region Adjacency Graph). Ce dernier se construit à partir de la segmentation obtenue par division de la manière suivante :

- A chaque région de l'image correspond à un nœud du graphe de contiguïté.
- Chaque couple de régions adjacentes dans l'image est représenté dans le graphe GCA par un lien qui connecte les deux nœuds correspondants.
- Chaque lien est associé à une valeur qui est une mesure de dissimilarité qui existe entre les deux régions qu'il relie.

Cette méthode est considérée comme une opération de post-traitement qui vient compléter un algorithme de segmentation.

1.3.2.4 Méthodes par division-fusion (Split and merge)

En se basant sur les deux concepts qu'on vient de citer à savoir la division et la fusion, Horwitz et Pavlidis ont développé une méthode mixte qui divise et fusionne simultanément les régions.

Cette méthode utilise un arbre quaternaire (Quad-tree) où chaque nœud représente une région. Si ce nœud n'est pas terminal, il possède au maximum quatre fils. Cette méthode se déroule comme suit :

- Division d'une région R_i en plusieurs sous régions disjointes, Si $P(R_i)=\text{faux}$.
- Fusion de deux régions adjacentes R_i et R_j si $P(R_i \cup R_j)=\text{vrai}$.
- Arrêt si le critère de fin de rassemblement est atteint.

L'exemple de la Figure (1.2) illustre ce principe et celui de la Figure (1.3) donne l'arbre quaternaire correspondante.

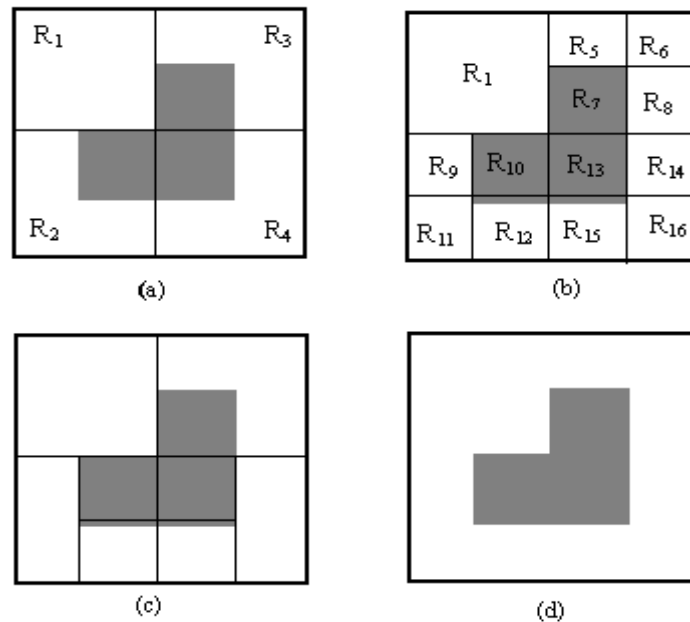


Figure (1.2) : Exemple d'application de l'algorithme de division –fusion

- (a) Image divisée en quatre régions, (b) image divisée en 13 régions,
(c) fusion de quelques régions et division de deux autres, (d) Image segmentée.

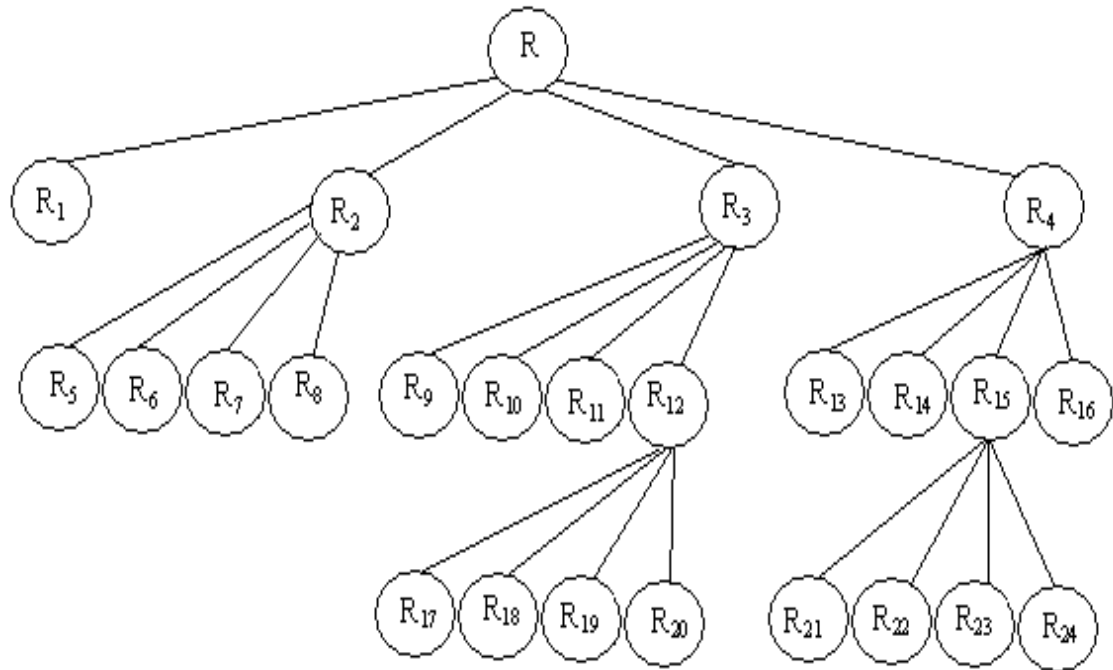


Figure (1.3) : Description du découpage de l'image au moyen d'un arbre quaternaire.

1.3.3 Approche par classification des pixels

Dans cette approche, les pixels sont caractérisés par un ensemble d'attributs. Ces attributs correspondent au niveau de gris. Les attributs de texture sont calculés à partir des pixels voisins situés à l'intérieure d'une fenêtre de voisinage centrée sur chaque pixels ou encore, aux composantes couleurs s'il s'agit d'une image couleur. Les pixels sont projetés dans l'espace des attributs et forment de nuages de points.

La classification des pixels consiste alors à extraire des nuages de points compacts qui correspondent aux classes de pixels dans l'image. Les méthodes de classification construisent les classes de pixels et affectent une étiquette à chaque pixel. La formation des régions homogènes dans l'image n'est obtenue qu'après l'analyse de connexité des pixels dans l'image étiquetée.

Les méthodes exploitant plusieurs attributs sont qualifiées de multidimensionnelles. Les méthodes ne prenant en compte qu'un seul attribut (niveau de gris) sont qualifiées de monodimensionnelles. Elles consistent à extraire automatiquement des seuils puis affectent les pixels à une classe par comparaison de leur niveau de gris à ces seuils. On les appelle communément méthodes de seuillage. Elles sont présentées dans le paragraphe suivant.

1.4 Segmentation d'image par seuillage

Le seuillage constitue une approche simple à la segmentation, elle est facile à mettre en œuvre et efficace dans les systèmes en temps réel. Ses performances sont cependant réduites car elles ne tirent pas profit de l'aspect spatial de l'information d'image. Elles sont

généralement recommandées lorsque les images présentent des classes évidentes : documents écrits ou schémas en noir et blanc ou en couleur, objets très contrastés (par exemple cellules d'une biopsie ou avion sur un ciel), etc.

Les techniques de seuillage sont basées sur l'hypothèse que les objets peuvent être distingués par leur niveau de gris.

L'objectif principal de ces méthodes est de déterminer d'une manière automatique le ou les seuils optimaux qui permettent de séparer les objets entre eux ou différents objets du fond.

1.4.1 Définition du seuillage

Les méthodes par seuillage consistent à affecter à chaque point d'image un numéro de classe, par comparaison des valeurs des niveaux de gris de ces points à des seuils déterminés préalablement en optimisant un critère donné.

Cette définition est formalisée de la manière suivante :

Soient $L=\{0, 1, 2, \dots, l-1\}$ l'ensemble de niveau de gris d'une image et $l(x, y)$ la luminance (niveau de gris) d'un pixel de coordonnées (x, y) . $l-1$ étant le niveau de gris maximal, souvent $l=256$.

La segmentation par seuillage est une opération qui consiste à répartir les pixels en K classes (C_1, C_2, \dots, C_k) à partir d'un ensemble de seuils $T=\{t_1, t_2, \dots, t_{k-1}\}$. Par convenance on utilise deux autres seuils, $t_0=0$ et $t_k=L-1$.

Un pixel de niveau de gris $l(x, y)$ est affecté à la classe C_k si :

$$t_k < l(x, y) \leq t_{k+1}, k=0, 1, 2, \dots, K-1$$

Le problème du seuillage consiste alors à déterminer les valeurs des seuils. Cependant, on distingue deux types de seuillage : seuillage local et seuillage global.

1.4.2 Seuillage local

On définit un seuil $t(x, y)$ pour chaque pixel de coordonnées (x, y) . Ce seuil est calculé en fonction des niveaux de gris des pixels situés autour du pixel étudié. Les techniques de seuillage local utilisent une fenêtre de voisinage centrée sur le pixel à étudier. Le premier à proposer une technique donnant de bons résultats fut Bernsen en 1986. Mathématiquement, le calcul du seuil est donné par :

$$t(x, y) = (\max(x, y) + \min(x, y)) / 2,$$

- $t(x, y)$: Seuil à appliquer pour le point (x, y)
- $\text{Max}(x, y)$: Valeur du niveau de gris maximal dans une fenêtre centré en (x, y) de taille $N*M$;

- $\text{Min}(x, y)$: Valeur du niveau de gris minimal dans une fenêtre centré en (x, y) de taille $N \times M$;

L'inconvénient de la méthode de seuillage local est sa sensibilité au bruit du fond.

1.4.3 Seuillage global

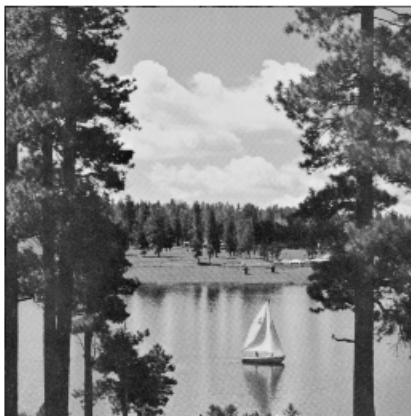
Un même seuil est affecté pour tous les pixels de l'image. Le calcul des seuils t_k ne dépend pas de la position des pixels, il repose sur l'exploitation de l'histogramme de l'image. Rappelons que l'histogramme représente la distribution des niveaux de gris ou la fréquence d'apparition des niveaux de gris dans une image. Il constitue une densité de probabilité de la variable liée à l'apparition du niveau de gris dans l'image. C'est un outil très privilégié en analyse de l'image. L'histogramme est décrit par une fonction discrète $h(i)$ ou $p(i)$ qui représente respectivement la fréquence et la probabilité d'apparition du niveau de gris i , tel que :

$$p(i) = \frac{h(i)}{N}$$

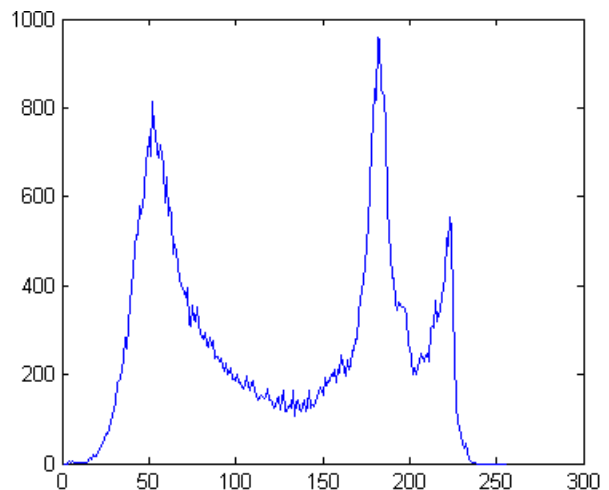
N étant le nombre total de pixels dans l'image.

Un histogramme peut être uni modal, il est dans ce cas formé d'un seul pic représentant un objet ou bien un fond. Un histogramme bimodal est caractérisé par deux modes séparés par une vallée, il indique l'existence d'un objet sur un fond. Un histogramme multimodal comporte plus de deux modes séparés par des vallées, ce qui correspond à une image comportant plusieurs régions ou objets (Figure 1.4).

Les valeurs des seuils recherchés sont souvent localisées dans les vallées de l'histogramme.



(a)



(b)

Figure (1.4) : Image réelle (a) et son histogramme (b).

En pratique, il est rare de trouver un histogramme qui présente deux modes bien distincts. En effet, la plupart des images présentent des histogrammes bruités caractérisés par des modes non discernables.

Plusieurs méthodes de seuillage ont été alors proposées.

1.4.3.1 Méthode d'Otsu

Cette méthode est sans doute la plus populaire. L'opération de seuillage est vue comme une séparation (un partitionnement) des pixels d'une image en deux classes C_0, C_1 (objet et fond) à partir d'un seuil t . Ces deux classes sont désignées en fonction du seuil t .

$$C_0 = \{0, 1, \dots, t\} \text{ et } C_1 = \{t+1, \dots, L-1\}$$

Soient σ_w^2 la variance d'une classe, σ_b^2 la variance interclasse et σ_T^2 la variance totale.

Le seuil optimum t^* peut être déterminé en maximisant un des trois critères suivant :

$$\lambda(t) = \frac{\sigma_b^2}{\sigma_w^2} ; \quad \eta(t) = \frac{\sigma_b^2}{\sigma_T^2} ; \quad \kappa(t) = \frac{\sigma_T^2}{\sigma_w^2}$$

$$\text{avec } \sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_t)^2 p_i ; \quad \mu_t = \sum_{i=0}^{L-1} i p_i ;$$

$$\sigma_b^2 = p_t q (\mu_1 - \mu_2)^2 ; \quad p_t = \sum_{i=0}^t \frac{h(i)}{N} ; \quad q = 1 - p_t ;$$

$$\mu_1 = \frac{\mu_t - \mu_s}{1 - p_t} ; \quad \mu_2 = \frac{\mu_s}{p_t} ; \quad \mu_s = \sum_{i=0}^t i p_i ;$$

Et $p_i = \frac{h(i)}{N}$: correspond à la probabilité d'apparition du niveau de gris i .

Les trois critères sont équivalents, mais le plus simple à utiliser est $\eta(t)$.

Le seuil optimum t^* est défini comme suit :

$$t^* = \underset{t \in L}{\text{Arg max}} \eta(t)$$

Cette méthode peut s'étendre à un nombre de classes plus important. Le principe est de choisir un certain nombre de seuils qui vont servir de séparateurs entre les objets qui comportent l'image. Dans ce cas, $(K-1)$ seuils optimaux $T = \{t_1, t_2, \dots, t_{k-1}\}$ peuvent être déterminés en minimisant la variance intra classes $\sigma_w^2(T)$ ou en maximisant la variance inter classes $\sigma_b^2(T)$ tel que :

$$\sigma_w^2(k) = \sum_{i=0}^{k-1} \sum_{j=t_i}^{t_{i+1}-1} (j - \mu_{i+1})^2 p_j , \quad \sigma_b^2 = \sum_{i=1}^k p_i (\mu_i - \mu_{i+1})^2$$

$$\sigma_T^2 = \sum_{j=0}^{L-1} (j - \mu)^2 p_j , \quad \sigma_T^2 = \sigma_w^2(k) - \sigma_b^2(k) , \quad \mu = \frac{1}{N} \sum_{j=0}^{L-1} j h_j ,$$

$$\mu_i = \frac{1}{N} \sum_{j=t_i}^{t_{i+1}-1} j h_j , \quad p_i = \frac{1}{N} \sum_{j=t_i}^{t_{i+1}-1} h_j$$

1.4.3.2 Méthode de seuillage basée sur l'entropie

Selon la théorie de l'information, l'entropie est une mesure de quantité d'information d'un système. Soit un ensemble fini $S=\{s_1, s_2, \dots, s_k\}$ d'événements indépendants, et p_i la probabilité d'occurrence de chaque élément s_i , l'entropie est définie par :

$$H(p_1, p_2, \dots, p_k) = - \sum_{i=1}^k p_i * \log (p_i)$$

Où : $\sum_{i=1}^k p_i = 1$

L'entropie d'une image peut être considérée comme une évaluation de l'information qu'elle contient. Elle permet également de déterminer le degré de désordre dans l'image. L'entropie est en général minimale pour une image homogène.

Plusieurs méthodes de seuillage utilisant la notion d'entropie ont été proposées. La plus connue d'entre elles est la méthode Kapur, Sahoo et Wang.

Dans cette méthode, deux distributions de probabilités, l'une relatif aux objets, l'autre au fond, découlent de la distribution des niveaux de gris de l'image originale, comme suit :

$$\frac{P_0}{P_t}, \frac{P_1}{P_t}, \dots, \frac{P_t}{P_t}, \frac{P_{t+1}}{1-P_t}, \dots, \frac{P_{l-1}}{1-P_t}$$

Où t est la valeur du seuil et $P_t = \sum_{i=0}^t p_i$

Soient l'entropie des deux distributions :

$$H_b = - \sum_{i=0}^t \frac{p_i}{p_t} \log \left(\frac{p_i}{p_t} \right) ; \quad H_n = - \sum_{i=t+1}^{L-1} \frac{p_i}{1-p_t} \log \left(\frac{p_i}{1-p_t} \right)$$

Le seuil optimal t^* est défini comme étant le niveau de gris qui maximise la quantité H_b+H_n : c'est-à-dire que :

$$t^* = Arg \max_{t \in L} \{H_b(t) + H_n(t)\}$$

Cette méthode peut être également étendue au calcul de plusieurs seuils. Il s'agira de maximiser le critère suivant :

$$J(T) = \sum_{i=1}^k H_i$$

avec $H_i = \sum_{j=t_{i-1}}^{t_i-1} \frac{p_j}{p_i} \log \left(\frac{p_j}{p_i} \right) ; \quad p_i = \frac{1}{N} \sum_{j=t_{i-1}}^{t_i-1} h_j$

1.5 Problème du multi-seuillage

Plusieurs méthodes de seuillage ont été développées pour déterminer un seul seuil. Ces méthodes peuvent être étendues au cas de multi seuillage comme cela a été décrit dans les paragraphes précédents.

Cependant, en pratique cette extension peut engendrer des temps de calculs prohibitifs. En effet, la recherche rapide d'un seul seuil d'une manière exhaustive est tout à fait possible. Cependant cette recherche exhaustive devient prohibitive lorsque le nombre de seuils augmente.

Chang et Al notent que la complexité des calculs augmente exponentiellement lorsque le nombre de seuils augmente. Pour K classes le nombre d'opérations NO nécessaires pour le calcul de $K-1$ seuils est donné par :

$$NO = \frac{(L + K)!}{L!K!}$$

Où L représente le nombre de niveau de gris.
Ainsi, pour une image ayant $L=256$ niveaux de gris :

$$NO=33153 \quad Si \quad K=2.$$

$$NO=2862209 \quad Si \quad K=3.$$

$$NO= 9711475137 \quad Si \quad K=5$$

Pour résoudre ce problème, plusieurs techniques ont été proposées. Parmi elles, on peut citer les méthodes d'optimisation métaheuristiques [Hammouche et al., 2010].

1.6 Conclusion

Dans ce chapitre, nous avons décrit brièvement les différentes approches et méthodes de segmentation. Nous avons aussi détaillé le seuillage d'histogrammes qui représente l'objet de notre travail. Nous avons ainsi remarqué que le problème du seuillage consiste à calculer la valeur d'un ou plusieurs seuils en optimisant un critère donné.

Nous avons également noté le problème de l'extension de la recherche des valeurs des seuils surtout lorsque leur nombre augmente. Nous proposons dans notre travail d'utiliser une méthode d'optimisation métaheuristique à base d'un algorithme à estimation de distribution pour résoudre le problème du seuillage. Quelques notions sur les AEDs sont présentées dans le chapitre suivant.

Chapitre 2

2.1 Introduction

Les algorithmes à estimation de distribution, en anglais Estimation of distribution algorithms (EDA) est une nouvelle classe des algorithmes évolutionnaires développée pendant les dix dernières années. Le principal avantage de ces algorithmes est l'absence des multiples paramètres de réglage et la présence des modèles de probabilité qui guident le processus de recherche.

Le premier AED a été proposé par Baluja en 1994 sous le nom « Population Based Incremental Learning », de nombreuses variantes de ces algorithmes ont été proposées depuis cette date et font objet d'une recherche active au sein de la communauté des algorithmes évolutionnaires et génétiques. Les premiers algorithmes se sont focalisés sur la représentation binaire du vecteur de solutions i.e la résolution des problèmes discrets. Plus tard, ils ont été modifiés pour pouvoir résoudre différents problèmes dans le domaine continu.

Les algorithmes à estimation de distribution utilisent les informations collectées tout au long du processus d'optimisation pour construire des modèles probabilistes des « bonnes régions » de l'espace de recherche et s'en servent pour engendrer de nouvelles solutions. Ces algorithmes sont caractérisés par le fait qu'ils n'utilisent pas d'opérateurs de croisement ni de mutation. La nouvelle population est engendrée en utilisant une distribution de probabilité estimée à partir des solutions de la génération précédente.

2.2 Description

Les algorithmes à estimation de distribution (EDA) évoluent en population de solutions potentielles dans un problème d'optimisation donné.

A l'instar des algorithmes évolutionnaires standards, les AEDs commencent par générer de façon aléatoire une population de solutions potentielles. A Chaque itération, on met à jour la population à l'aide d'opérateurs de sélection et de variation. L'opérateur de sélection sélectionne une population de solutions prometteuses à partir de la population actuelle. Cependant, au lieu d'utiliser les opérateurs de variation inspirée de la nature de l'évolution et de la génétique, les AEDs créent de nouvelles solutions en construisant un modèle probabiliste de solutions sélectionnées et l'échantillonnage du modèle construit permet de générer de nouvelles solutions. Les nouvelles solutions sont incorporées dans la population originale et le processus de sélection et de variation est itéré jusqu'à satisfaire un critère d'arrêt. Idéalement, la qualité des solutions générées par le modèle probabiliste s'améliore au fil du temps, et, après un nombre raisonnable d'itérations, l'échantillonnage du modèle devrait générer seulement le meilleur optimum. Dans cette mise en œuvre, le modèle probabiliste construit pour la population de solutions sélectionnées forme un arbre de dépendance où chaque variable est conditionnée à son prédécesseur. La distribution globale est ensuite donnée comme le produit de la probabilité marginale de la variable à la racine de l'arbre et les probabilités conditionnelles des variables restantes compte tenu de leurs prédécesseurs. Utilisation des arbres probabilistes est une approche commune dans la machine d'apprentissage parce que l'apprentissage des modèles généraux, tels que les

réseaux bayésiens de connexions multiples, est parfois insolubles. Les modèles probabilistes d'arbres ont également été utilisés dans les AEDs.

2.3 Les différentes étapes de L'AED

Tout d'abord, on génère un ensemble de solutions (individus) aléatoirement suivant par exemple une distribution uniforme. Ces solutions seront évaluées à l'aide d'une fonction objectif. Cette fonction objectif évalue l'influence de chacune des solutions au problème à optimiser. A base de cette évaluation, un nombre de solutions sera sélectionné (les solutions possédant de bonnes valeurs de la fonction objectif). Ensuite, un modèle probabiliste des solutions sélectionnées sera construit et une population de solutions est générée à partir de ce modèle probabiliste. Le processus est itéré jusqu'à l'obtention de l'optimum.

Les étapes de l'algorithme à estimation de distribution sont les suivantes :

1. On génère une population D_0 de N solutions aléatoirement suivant une distribution uniforme
2. Evaluation des N solutions générées à l'aide d'une fonction objectif
3. Sélectionner M solutions (individus) à partir des N solutions. ces solutions sélectionnées forment la population : D_1^M
4. Estimer les paramètres du modèle à partir des M individus (construction du modèle probabiliste à n dimensions $p_1(x/D_1^M)$)
5. Générer N nouvelles solutions (D_{1+1}) à partir du modèle probabiliste $P(x/D_1^M)$ (Echantillonnage)
6. Reprendre les étapes 2,3,4,5 jusqu'à ce que le critère d'arrêt soit satisfait

Chaque solution est un vecteur de valeurs considérées comme des variables statistiques. Le modèle graphique probabiliste est construit à partir des individus sélectionnés et les relations entre les différentes variables qui forment chaque individu sont exprimées explicitement à partir des distributions probabilité jointes associées aux individus sélectionnées à chaque itération.

L'étape d'estimation de la distribution de probabilité (construction du modèle probabiliste) associée aux données des individus sélectionnés est la plus difficile à réaliser dans cet algorithme.

Suivant le modèle probabiliste choisi, on peut trouver trois différentes variantes de ces algorithmes :

- Les algorithmes utilisant le modèle monovariante
- Les algorithmes utilisant le modèle bivariante
- Les algorithmes utilisant le modèle multivariante

Le choix du modèle probabiliste est relatif à la dimension du problème à résoudre (nombre de variables) et le type de représentation (discret, continu...).

2.4 Rappels sur les distributions de probabilités jointes

Soit X_i une variable aléatoire et x_i est l'une de ses valeurs possibles. Soit $p(X_i = x_i)$ (ou simplement $p(x_i)$) la fonction de densité de probabilité au point x_i . $X = \{X_1, X_2, \dots, X_n\}$ est le vecteur des n variables et $x = \{x_1, x_2, \dots, x_n\}$ est la valeur de chacune des variables du vecteur X et $\rho(X = x)$ (ou simplement $p(x)$) représente la fonction de densité de probabilité jointe de X . De la même façon, la fonction de densité de probabilité conditionnelle de la variable X_i de valeur x_i sachant que la variable X_j a pour valeur x_j sera représentée par : $\rho(X_i = x_i \setminus X_j = x_j)$ (ou simplement $p(x_i \setminus x_j)$).

Si le problème à traiter appartient au domaine discret, i.e la variable X_j est discrète, $\rho(X_i = x_i) = p(X_i = x_i)$ (ou simplement $p(x_i)$) est la distribution marginale de la variable X_i . si toutes les variables dans X sont discrètes, alors $\rho(X = x) = p(X = x)$ (ou simplement $p(x)$) sera la distribution de probabilité jointe complète. De la même manière, la probabilité conditionnelle que X_i soit égale à une valeur x_i sachant que X_j est égale à x_j sera notée par : $\rho(X_i = x_i \setminus X_j = x_j)$ (ou simplement $p(x_i \setminus x_j)$).

Dans le domaine continu, X_j est une variable continue, $\rho(X_i = x_i) = f(X_i = x_i)$ (ou simplement $f(x_i)$) est la fonction densité de la variable X_i . Si toutes les variables dans X sont continues, alors $\rho(X=x)=f(X=x)$ (ou simplement $f(x)$) sera la densité jointe. De la même manière, $\rho(X_i = x_i \setminus X_j = x_j) = f(X_i = x_i \setminus X_j = x_j)$ (ou simplement $f(x_i \setminus x_j)$) sera la densité conditionnelle de X_i de valeur x_i sachant que X_j a pour valeur x_j .

On peut définir la distribution marginale de la $i^{\text{ème}}$ variable par $p(X_i = x_i)$ (ou simplement $p(x_i)$), aussi la somme des probabilité de tous les vecteurs d'une population ayant $X_i = x_i$ et on l'écrit sous cette forme :

$$p(x_i) = \sum_{X, X_i=x_i} P(X)$$

Soit X_s un sous vecteur de X et x_s est l'ensemble des valeurs possible prises par X_s , alors la probabilité marginale $p(X_s = x_s)$ (ou simplement $p(x_s)$) est définie comme une somme de probabilité de tous les vecteurs d'une population ayant le même ensemble de valeurs x_s du sous vecteur X_s et on peut l'écrire sous cette forme :

$$p(x_s) = \sum_{X, X_s=x_s} P(X)$$

Soit X_A et X_B deux sous vecteurs disjoints de X et x_A, x_B deux sous ensemble de valeurs possibles correspondantes respectivement aux deux vecteurs, alors la probabilité conditionnelle $p(X_A = x_A \setminus X_B = x_B)$ (ou simplement $p(x_A \setminus x_B)$) peut être calculée comme suit :

$$P(x_A \setminus x_B) = \frac{p(x_A, x_B)}{p(x_B)}$$

D'après la théorie des probabilités, la distribution de probabilité jointe complète d'un groupe d'individus $P_r(X=x)$ ou simplement $P_r(x)$ où chaque variable $X=\{X_1, X_2, \dots, X_n\}$ prend les valeurs $x=\{x_1, x_2, \dots, x_n\}$ peut être calculée comme suit :

$$P_r(x) = p(x_1 \setminus x_2, \dots, x_n) p(x_2 \setminus x_3, \dots, x_n) \dots p(x_{n-1} \setminus x_n) p(x_n)$$

2.4.1 Les différentes lois de probabilités

Il existe plusieurs types de fonctions de densité de probabilité décrivant les différentes lois probabilistes. Dans le cas de variables continues, on peut citer :

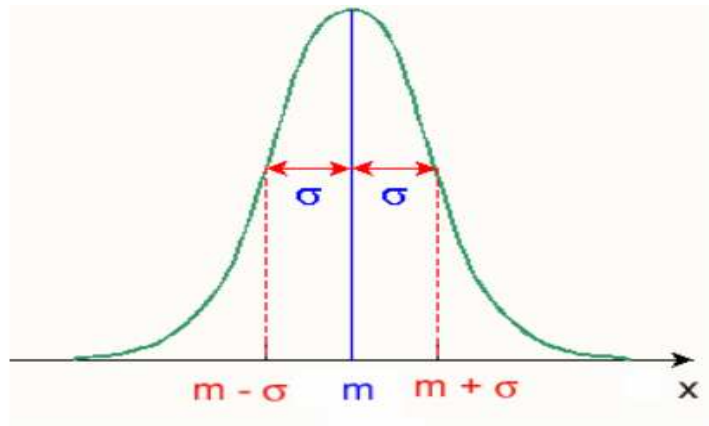
➤ Loi Normale

La distribution normale est une distribution théorique, en ce sens qu'elle est une idéalisation mathématique qui ne se rencontre jamais exactement dans la nature. Mais de nombreuses distributions réellement observées s'en rapprochent et ont cette fameuse forme de « cloche » (beaucoup d'individus autour de la moyenne, de moins en moins au fur à mesure qu'on s'en éloigne, et ceci de façon symétrique). Elle a été introduite par le mathématicien Abraham de Moivre en 1733 afin d'approcher des probabilités associées à des variables aléatoires binomiales possédant un paramètre n très grand. Cette loi a été mise en évidence par Gauss au XIX^e siècle et permet de modéliser de nombreuses études biométriques. Sa densité de probabilité a la forme d'une courbe dite courbe en cloche ou courbe de Gauss.

La loi normale de paramètres m et σ , notée $N(m, \sigma)$, est définie sur \mathbb{R} par la densité :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2}$$

dont la représentation graphique est la suivante :



Notons que : la droite $x = m$ est axe de symétrie

les points d'inflexion sont situés à une distance σ de cet axe de symétrie

➤ Loi bêta

La loi bêta est une famille de lois de probabilités continues, définies sur $[0,1]$, paramétrée par deux paramètres de forme, typiquement notés α et β .

Sa densité de probabilité est donnée par :

$$f(x) = \begin{cases} \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$$

où : B est la fonction beta.

$$\text{avec : } E(x) = \frac{\alpha}{\alpha + \beta}, \text{ si } \beta > 1, \text{ var}(x) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \text{ si } \beta > 2$$

où : $E(x)$ et $\text{var}(x)$ sont respectivement l'espérance et la variance.

➤ Loi Gamma

Une distribution Gamma, ou loi Gamma, est un type de loi de probabilité de variables aléatoires réelles positives. La famille des distributions Gamma inclut entre autres les lois exponentielles, les lois de sommes de variables aléatoires indépendantes suivant une même loi exponentielle. Elle permet donc de modéliser une grande variété de phénomènes pour des grandeurs positives.

Une variable aléatoire X suit une loi Gamma de paramètres k et θ (strictement positifs), notée $\Gamma(k, \theta)$, si sa fonction de densité de probabilité peut se mettre sous la forme :

$$f(x; k; \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\Gamma(k)\theta^k}$$

avec : $E(x) = k\theta$, $var(x) = k\theta^2$

où : $E(x)$ et $var(x)$ représentent respectivement l'espérance et la variance.

➤ La loi log-normale

Une variable aléatoire X est dite suivre une loi log-normale de paramètres μ (pour l'espérance) et σ^2 (pour la variance) si le Log de la variable suit une loi normale de paramètres μ (pour l'espérance) et σ^2 (pour la variance).

Cette loi est parfois appelée loi de Galton.

La loi log-normale de paramètres μ et σ admet pour densité :

$$f(x, \mu, \sigma) = \frac{e^{-0.5\left(\frac{\ln x - \mu}{\sigma}\right)^2}}{x\sigma\sqrt{2\pi}}$$

Pour $x > 0$. μ et σ sont la moyenne et l'écart type du logarithme de la variable (puisque par définition, le logarithme de la variable est distribué selon une loi normale de moyenne μ et d'écart-type σ).

➤ Loi du Khi-deux

La loi du Khi-deux à n degrés de liberté, notée X_n^2 , avec n un entier positif, a une fonction de densité pour $x > 0$:

$$f(x, n) = \frac{1}{2^{n/2}\Gamma(n/2)} e^{-x/2} x^{\frac{n}{2}-1}$$

Ses moments se déduisent de la loi gamma :

$$E(X_n^2) = \frac{n/2}{1/2} = n \text{ et } v(X_n^2) = \frac{n/2}{1/4} = 2n$$

➤ **Loi de pareto**

C'est une loi qui est utilisée notamment dans la modélisation de la distribution des revenus d'une population ou en théorie des assurances. Sa densité est définie pour $x \geq x_0 > 0$, x_0 pouvant s'interpréter comme le revenu minimum, en fonction d'un paramètre $\alpha > 0$:

$$f(x) = \frac{\alpha}{x_0} \left(\frac{x_0}{x}\right)^{\alpha+1}$$

➤ **Loi de Weibull**

Une variable aléatoire continue suit une loi de Weibull, notée $W(k, \lambda)$ de paramètre $k > 0$ et $\lambda > 0$, si sa fonction de densité de probabilité est :

$$f(x, k, \lambda) = (k/\lambda) (x/\lambda)^{(k-1)} e^{-(x/\lambda)^k}$$

où k est le paramètre de forme et λ est le paramètre d'échelle.

➤ **Loi de Rayleigh**

Une variable aléatoire continue suit une loi de Rayleigh $R(\sigma)$ de paramètre $\sigma > 0$, si sa fonction densité prend la forme suivante :

$$f(x) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right), x \in [0; +\infty[$$

➤ **Loi exponentielle**

Une variable aléatoire continue X suit une loi exponentielle $E(\mu)$ de paramètre $\mu \in \mathbb{R}$, si sa fonction densité s'exprime par :

$$f(x, \mu) = \frac{1}{\mu} e^{-\frac{x}{\mu}}$$

On dit aussi que X suit une loi exponentielle de paramètre $\lambda = \frac{1}{\mu}$ telle que la fonction de densité $E(\lambda)$ est :

$$f(x, \lambda) = \lambda e^{-\lambda x}$$

2.5 Les différents AEDs

2.5.1 Modèle monovariante

Les algorithmes à estimation de distribution de cette catégorie ne considèrent aucune dépendance entre les variables d'un individu. La distribution de probabilité jointe devient un produit simple des probabilités marginales mono variable de tous les variables d'un individu.

$$P(x) = \prod_{i=1}^n p(x_i)$$

Où n est le nombre total de variable d'un individu et $p(x_i)$: probabilité marginale de la $i^{\text{ème}}$ variable.

Les algorithmes de cette catégorie sont simples et efficaces lorsque les problèmes sont linéaires (une fonction d'optimisation où les variables ne sont pas interdépendantes). Cependant, ils sont peu performants lors de la résolution des problèmes complexes où les variables sont dépendantes.

2.5.1.1 Algorithme de distribution marginale monovariante

Cet algorithme à distribution marginale monovariante, connu sous le nom de UMDA (Univariate marginal distribution algorithm) est le premier algorithme à estimation de distribution proposé par [Mühlenbein & Paaß (1996)]. Il suit les étapes suivantes :

- 1 Générer aléatoirement une population initiale de taille N
- 2 Sélectionner M solutions (individus), $M < N$
- 3 Calculer la probabilité marginale monovariante $P(x_i)$ à partir des solutions sélectionnées
- 4 Remplacer la population précédente par les N nouvelles solutions générées à partir des distributions marginales

$$P(x) = \prod_{i=1}^n p(x_i)$$

- 5 Retourner à l'étape 2 jusqu'à ce que le critère d'arrêt soit satisfait

2.5.1.2 La population basée sur l'apprentissage incrémental (PBIL)

L'algorithme PBIL a été proposé par [Baluja (1994)]. Il est caractérisé par la représentation binaire d'un individu. La première étape de l'algorithme consiste à initialiser un vecteur de probabilité, le maintenir. Et au cours du processus de calcul, le vecteur de probabilité est mis à jour.

Les étapes de cet algorithme sont les suivantes :

- 1) Initialiser le vecteur de probabilité $\{ p_1, p_2, \dots, p_n \}$ avec la valeur de 0,5 pour chaque probabilité
- 2) Générer $M \gg 0$ individus selon les probabilités p_i puis les évaluer.
- 3) Mise à jour du vecteur de probabilité en fonction du plus fort individu $S = \{ s_1, s_2, \dots, s_n \}$ en utilisant la règle suivante:

$$p_i = p_i * (1.0 - LR) + s_i * LR$$

LR est taux d'apprentissage, choisi préalablement.

- 4) Si la condition de mutation est valide, la mutation du vecteur de probabilité utilise la règle suivante :

$$p_i = p_i * (1.0 - MS) + random(0 \text{ ou } 1) * MS$$

MS est taux de mutation

- 5) Passer à l'étape 2 jusqu'à ce que le critère de terminaison soit satisfait.

2.5.1.3 Algorithme génétique compact (CGA)

L'algorithme génétique compact (CGA) a été proposé par [Harik, Lobo & Goldberg (1998)]. Cet algorithme utilise également un vecteur de probabilité comme dans PBIL. Cependant, contrairement à PBIL, l'algorithme génétique compact échantillonne seulement deux individus à la fois.

Les étapes de cet algorithme sont les suivantes:

- 1) Initialiser le vecteur de probabilité $\{ p_1, p_2, \dots, p_n \}$ avec la valeur 0,5 pour chaque probabilité
- 2) Echantillonner 2 individus en fonction des probabilités p et de les étiquetter en fonction de leur aptitude: gagnant pour le plus fort et perdant pour le moins fort
- 3) Mise à jour du vecteur de probabilités en utilisant la règle suivante:
 - Pour $i = 1$ à n faire
 - Si le gagnant $[i] \neq$ perdant $[i]$, alors
 - Si le gagnant $[i] = 1$ alors $p_i = p_i + \text{updaterate}$
 - Sinon $p_i = p_i - \text{updaterate}$
 - Où updaterate est une petite valeur, généralement, définie comme $1/(\text{la taille de la population})$.
- 4) Passer à l'étape 2 jusqu'à ce que le vecteur de probabilité converge.

2.5.2 Modèle bivarié

Les algorithmes de cette catégorie considèrent une dépendance d'ordre deux entre les variables d'un individu. Le modèle probabiliste devient plus complexe comparativement au modèle monovarié. Ces algorithmes sont efficaces lors de la résolution de problèmes possédant une interaction double entre les variables. Cependant, ils présentent un échec dans le cas des problèmes possédant des variables à interactions multiples. Mutual Information Maximization for input clustering (MIMIC), Combining Optimizers with Mutual Information Trees (COMIT), Bi-variate Marginal Distribution Algorithm (BMDA) sont des exemples de L'AED utilisant le modèle bivarié de distribution de probabilité.

La figure ci-dessous donne une représentation graphique du modèle probabiliste bivarié :

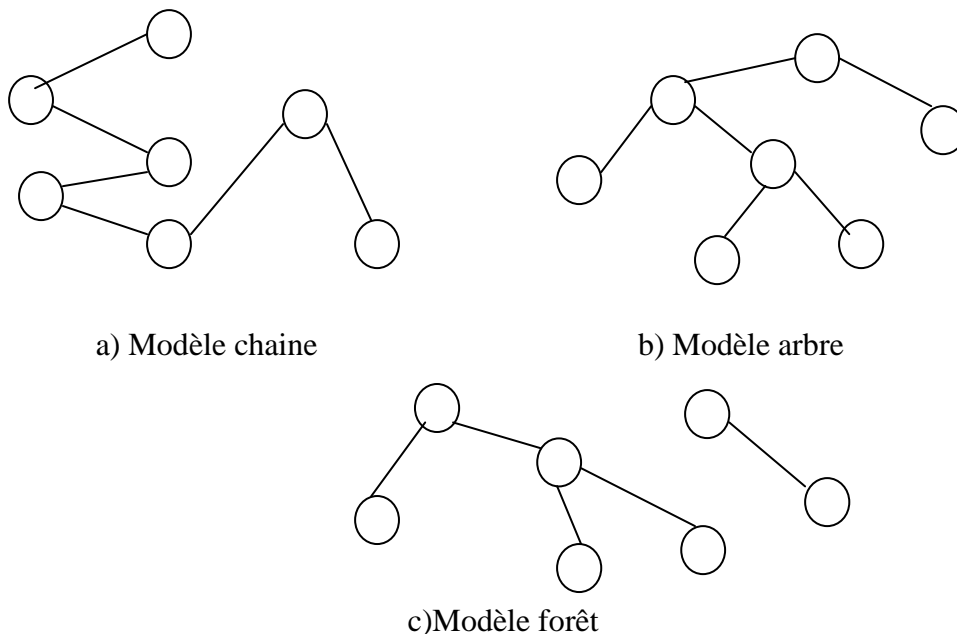


Figure (2.1) : Représentation graphique du modèle de probabilité en supposant la dépendance d'ordre deux entre les variables.

2.5.2.1 Maximisation de l'information mutuelle

L'algorithme MIMIC a été proposé par [De Bonet, Isbell and Viola (1997)]. Il utilise une chaîne structurée de la distribution de probabilité qui peut s'écrire sous cette forme :

$$P_{\pi}(x) = p(x_{i_1} \setminus x_{i_2}) p(x_{i_2} \setminus x_{i_3}) \dots \dots \dots p(x_{i_{n-1}} \setminus x_{i_n}) p(x_{i_n})$$

Où : $\pi = i_1 i_2 \dots i_n$ est une permutation de nombres entre $1 \dots n$. La distribution $P_{\pi}(x)$ utilise π comme un ordre pour la probabilité conditionnelle paire. Le but est de trouver l'ordre π de telle manière que la distribution $P_{\pi}(x)$ soit égale à la distribution de probabilité jointe complète $P_r(x)$.

$$P_r(x) = p(x_1 \setminus x_2 \dots x_n) p(x_2 \setminus x_3 \dots x_n) \dots p(x_{n-1} \setminus x_n) p(x_n).$$

La divergence Kullback-Liber $D(P_r(x) \setminus P\pi(x))$ est utilisée pour mesurer l'identité entre $P_r(x)$ et $P\pi(x)$.

2.5.2.2 Combinaison des Optimiseurs

L'algorithme COMIT (en anglais Combining Optimizers with Mutual Information Trees (COMIT) a été proposé par [Baluja & Davies (1997)], il utilise aussi le modèle de distribution bivariable et la structure de cette distribution est donnée sous forme d'un arbre. COMIT utilise l'algorithme Maximum Weight Spanning Tree (MWST) pour construire une structure d'arbre, elle utilise aussi sa probabilité qui peut s'écrire sous cette forme :

$$P(x) = \prod_{i=1}^n p(x_i \setminus x_j)$$

Où : j représente la position parent pour i. Cependant, quand un parent pour la i^{ème} variable n'existe pas, alors $p(x_i \setminus x_j)$ va se mettre sous cette forme : $p(x_i)$.

2.5.2.3 Algorithme de distribution marginale bivariable

L'Algorithme de distribution marginale bivariable a été proposé par [Pelikan & Mühlenbein (1999)]. Cet algorithme est une extension de l'algorithme de distribution marginale monovariable. Il peut résoudre deux classes de problèmes : les problèmes linéaires et les problèmes possédant une double interaction entre les variables.

Pour les structures de chaîne et d'arbre, l'algorithme de distribution marginale bivariable utilise la structure forêt pour représenter le modèle de probabilité.

La dépendance entre les variables est détectée durant le processus d'optimisation.

L'algorithme de distribution marginale bivariable utilise les statistiques Pearson's chi-2 pour la dépendance.

Les étapes de l'algorithme sont les suivantes :

- Générer aléatoirement une population initiale de taille N
- Sélectionner M solutions, tel que $M < N$
- Calculer la probabilité marginale monovariable $p(x_i)$ et la probabilité marginale bivariable $p(x_i, x_j)$ des solutions sélectionnées
- Construction du modèle probabiliste par la mesure de la dépendance entre chaque paire de solutions en utilisant les statistiques de Pearson chi-2

- Remplacer les K solutions parents par les K nouvelles solutions générées selon le modèle de probabilité construit
- Retour à l'étape 2 jusqu'à ce que le critère d'arrêt soit satisfait

2.5.3 Modèle multivariable

Les algorithmes qui considèrent l'interdépendance entre variables d'ordre supérieur à deux peuvent appartenir à cette classe. L'Algorithme génétique compact étendu, l'algorithme de distribution factorisée (FDA), Algorithme d'optimisation bayésienne (BOA) sont des exemples de L'AED utilisant le modèle multivariable de distribution de probabilité.

La figure ci-dessous montre la représentation graphique de différents réseaux probabilistes utilisés par ces algorithmes.

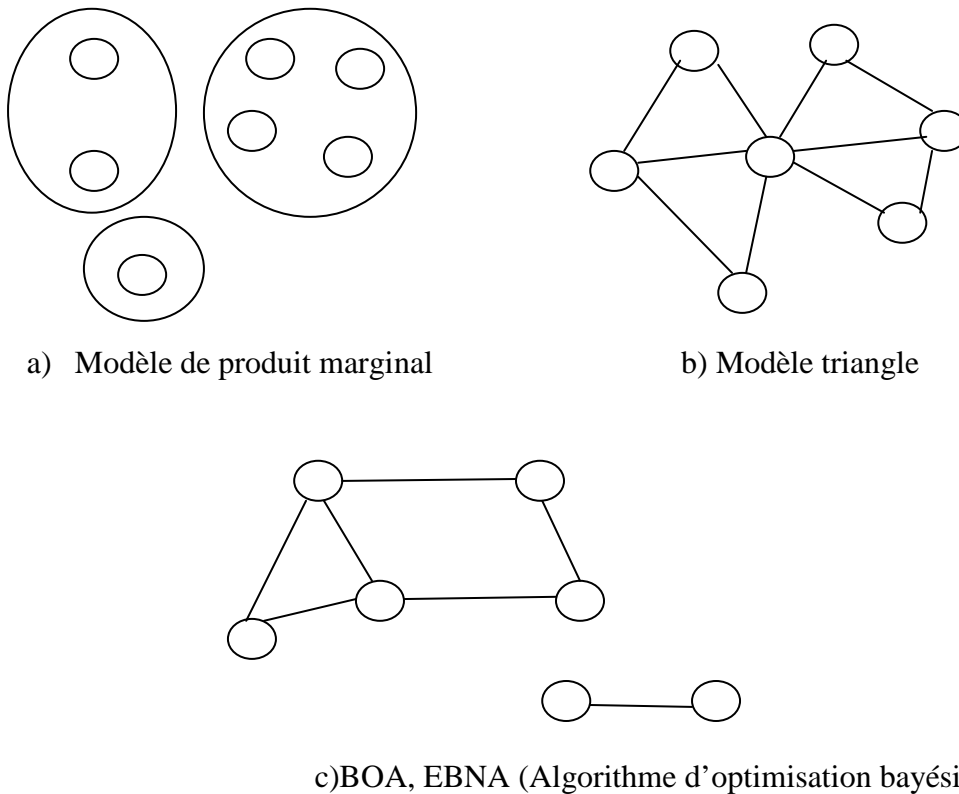


Figure (2.2): Représentation graphique du modèle de probabilité multivariable

2.5.3.1 Algorithme génétique compact étendu (ECGA)

L'algorithme génétique compact étendu (ECGA) a été proposé par Harik en 1999. Le modèle de probabilité utilisé dans cet algorithme s'écrit sous cette forme :

$$p(x) = \sum_{c \in C} p(x_c)$$

où, la probabilité marginale d'un groupe de c variables dépendantes est représentée par $p(x_c)$.

2.5.3.2 Algorithme de distribution factorisée (FDA)

L'algorithme de distribution factorisée a été proposé par Mühlenbein & Mahnig en 1999. Les étapes de cet algorithme sont les suivantes :

- 1- En utilisant la décomposition additive de la fonction calculant b_i (résidus) et c_i (séparateurs): l'ensembles de variables sont corrélées les unes aux autres
- 2- Générer N individus selon la distribution uniforme :

$$P(x) = \prod_{i=1}^n p(x_i)$$

- 3- Sélectionner $M \leq N$ individus en utilisant la sélection de Boltzmann où la probabilité pour qu'un individu X soit sélectionné est calculé comme suit :

$$P_{\beta}(x) = \prod_{i=1}^k P_{\beta}(x_{b_i} \setminus x_{c_i}) = \frac{\prod_{i=1}^k P_{\beta}(x_{b_i}, x_{c_i})}{\prod_{i=2}^k P_{\beta}(x_{c_i})}$$

où k est le nombre de l'ensemble des variables corrélées;
 P_{β} est la Distribution de Boltzmann calculée comme suit :

$$P_{\beta}(\alpha) = \frac{e^{-\frac{f(\alpha)}{t}}}{Z} \quad \text{où :}$$

$Z = \sum_b e^{-\frac{f(b)}{t}}$ est la fonction de partition, $f(\alpha)$ est la fitness d'un individu α et T est la valeur de la température, choisie de façon que la distribution de Boltzmann soit appropriée pour les problèmes à optimiser.

- 4- Estimer la probabilité conditionnelle des individus sélectionnés.
- 5- Générer M nouvel individus selon la distribution conditionnelle

$$P(x) = \prod_{i=1}^n P(x_{b_i} \setminus x_{c_i})$$

- 6- Retour à l'étape 3 jusqu'à la satisfaction du critère

2.5.3.3 Algorithme d'optimisation Bayésienne (BOA)

L'algorithme d'optimisation bayésienne (BOA) a été proposé par Pelikan, Goldberg & Cantú-Paz en 1999. Dans cet algorithme la distribution de probabilité conjointe est encodée sous forme de réseau bayésien et le réseau bayésien se construit à partir des solutions sélectionnées.

Les différentes étapes de cet algorithme sont les suivantes :

- 1- Générer au hasard une population initiale de taille $N \gg 0$
- 2- Sélectionnez M solutions prometteuses où $M \leq N$
- 3- Construire un réseau bayésien B en utilisant une métrique choisie et k contraintes
- 4- Générer K nouvelles solutions en fonction de la distribution conjointe encodée par le réseau bayésien construit B
- 5- Créer une nouvelle population en remplaçant les k solutions parents par K nouvelles solutions générées.
- 6- Passer à l'étape 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Le tableau suivant donne les différentes variantes de L'AED [R.Santana et al.2008]

Ordre statistique	Avantages	Inconvénients	Exemples
Monovariante (Statistiques d'ordre un)	-Simple et rapide -Convient aux problèmes de cardinalité élevée -Evolutif	-Ignore les caractéristiques de dépendance -Mauvaises performances pour les problèmes d'ordre supérieur à 1.	PBIL (Baluja, 1994) UMDA (Mühlenbein and Paaß, 1996) CGA (Harik et al., 1999)
Bivariable (statistiques d'ordre deux)	-Capable de représenter un ordre de faible dépendance. -Convient à de nombreux problèmes.	-Une caractéristique de dépendance peut être ignorée. -Plus lent que L'AED monovariante.	MIMIC (De Bonet et al., 1996) arbres de dépendance EDA (Baluja et Davies, 1997) BMDA (Pelikan et Mühlenbein, 1999) Tree-EDA/Mixture des distributions EDA (Santana et al., 1999)
Multivariable (Statistiques d'ordre supérieur à deux)	-Apprentissage de paramètres (seulement les paramètres du modèle d'interaction) -Convient à des problèmes dont on sait le modèle sous-jacent -Puissance maximale de la généralisation -Flexibilité d'introduire l'utilisateur de dépendances	-Les caractéristiques de dépendance complexes peuvent être ignorées. -Ca nécessite une mémoire plus élevée la bi variée. -Temps de calcul plus élevé. -Plus hautes exigences mémoire.	FDA (Mühlenbein et al., 1999) Markov basé sur le réseau EDA (Shakya et McCall, 2007) -ECGA (Harik et al., 1999) EBNA (Ettxeberria et Larrañaga, 1999) BOA / hBOA (Pelikan et al., 1999, 2005) Dépendance réseaux EDA (Gámez et al., 2007)

Tableau (2.1) : Comparaison des différentes variantes de L'AED

2.6 Conclusion

Dans ce chapitre, nous avons présenté quelques notions sur les algorithmes à estimation de distribution ainsi que ses différentes variantes.

Nous avons constaté qu'il existe un très grand nombre d'algorithmes à estimation de distribution dans la littérature, ce qui prouve la richesse de ces algorithmes et l'intérêt qu'on leur porte

Dans le chapitre suivant, nous allons montrer comment résoudre le problème du seuillage en utilisant l'algorithme à estimation de distribution.

Chapitre 3

3.1 Introduction

Dans ce chapitre, nous allons appliquer l'algorithme à estimation de distribution au seuillage d'histogrammes.

Cette application a été réalisée sous MATLAB 7.8.0 sur un ordinateur doté d'un microprocesseur de type Intel (i3) à 2.40 Ghz/s et 4 Gbits de mémoire vive.

3.2 Seuillage d'histogramme à base de L'AED

Le problème du seuillage consiste à calculer la valeur d'un ou plusieurs seuils en optimisant un critère donné. La recherche d'un seul seuil peut se faire facilement et rapidement d'une manière exhaustive. Cependant cette recherche exhaustive devient prohibitive lorsque le nombre de seuils augmente. Nous avons ainsi proposé d'appliquer l'algorithme à estimation de distribution au seuillage d'histogramme pour rechercher rapidement les seuils.

L'AED utilisé pour résoudre le problème du seuillage est conçu sur le modèle UMDA (Univariate Marginal Distribution Algorithm).

Dans notre application, les seuils de l'histogramme représentent les variables aléatoires et la fonction d'Otsu représente la fonction objective à évaluer.

Les distributions marginales utilisées sont : La loi Normale, La loi Log-Normale, La loi Gamma, La loi Weibull, La loi Rayleigh et La loi exponentielle.

Et les étapes de l'algorithme à estimation de distribution sont les suivantes:

1. Initialiser une population $D^{(0)}$ de N solutions aléatoirement suivant une loi uniforme
2. $t=0$
3. Evaluer la population $D_N^{(t)}$ (calcul de la fonction objective)
4. Sélectionner M individus à partir de la population : $D_M^{(t)}$
5. Estimer les paramètres de la fonction densité de probabilité dont le modèle est connu à partir des M individus
6. Générer N nouvelles solutions à partir de $p(x/D_M^{(t)})$ (Echantillonnage)
7. Aller à l'étape 3 si le critère d'arrêt n'est pas satisfait

Dans le cas du seuillage d'histogrammes à base de L'AED, nous allons initialiser aléatoirement une population de seuils dont les valeurs sont comprises entre 0 et 255 et de taille =N*Nombre de seuils. Ensuite, nous allons évaluer chaque individu de la population initiale en calculant la fonction objective d'Otsu. A partir de cette évaluation, nous allons procéder à la sélection des M meilleurs individus à partir de la population initiale des N individus. Une fois la sélection est achevée, nous allons choisir un modèle probabiliste pour chaque distribution marginale et nous allons estimer les paramètres de la distribution. Les paramètres estimés vont nous permettre de générer N nouvelles solutions c'est-à-dire effectuer

l'opération d'échantillonnage. Le processus itératif ne s'arrête jusqu'à ce que le critère d'arrêt soit satisfait.

Les différents paramètres et fonctions de densité de probabilité des lois de distribution marginales sont résumés dans le tableau ci-dessous :

Distribution	Fonction de densité de probabilité	Moyenne $E(X)$	Variance $V(X)$
Normale $N(\mu, \sigma^2)$	$f(x) = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right], x \in R$	M	σ^2
Gamma $\gamma(\theta, k)$	$f(x) = \frac{\theta^k}{\Gamma(k)} x^{k-1} e^{-\theta x}, x > 0$ Où $\Gamma(k) = \int_0^{+\infty} e^{-x} x^{k-1} dx$	$\frac{k}{\theta}$	$\frac{k}{\theta^2}$
Log Normal $L(\mu, \sigma)$	$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{\ln x - \mu}{\sigma} \right)^2 \right\}, x > 0$	$\left(\mu + \frac{\sigma^2}{2} \right)$	$[e^{\sigma^2} - 1] e^{2\mu + \sigma^2}$
Exponentielle $E(\lambda)$	$f(x) = \lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Weibull $W(k, \lambda)$	$f(x; k; \lambda) = (k/\lambda)(x/\lambda)^{k-1} e^{-(x/\lambda)^k}$	$\lambda \Gamma \left(1 + \frac{1}{k} \right)$	$\lambda^2 \Gamma \left(1 + \frac{2}{k} \right) - E^2(X)$
Rayleigh $R(\sigma)$	$f(x; \sigma) = \frac{x}{\sigma^2} \exp \left(\frac{-x^2}{2\sigma^2} \right), x \geq 0$	$\sigma \sqrt{\frac{\pi}{2}}$	$\frac{4 - \pi}{2} \sigma^2$

Tableau (3.1): Paramètres et fonctions de densités de probabilités des 6 distributions marginales

3.3 Tests et résultats

Nous allons présenter dans ce paragraphe les résultats de la segmentation d'image par seuillage d'histogrammes basé sur l'algorithme à estimation de distribution décrit précédemment.

Les résultats seront donnés en faisant varier le nombre de seuil de 1 à 4 et le modèle probabiliste. 6 modèles ont été testés, il s'agit des lois Normale, Log-Normale, Gamma, Weibull, Rayleigh et exponentielle.

L'AED est exécuté avec les paramètres suivants :

- Taille de la population N=300
- Taille de la population sélectionnée M=100
- Nombre maximal d'itérations=20

Les valeurs de ces paramètres ont été déterminés expérimentalement et après plusieurs tests sur différentes images.

Cependant, nous allons présenter les résultats de seuillage obtenus sur les deux images de la figure (3.1). Les histogrammes de ces deux images sont affichés sur la figure (3.2).

Les résultats comprennent les valeurs des seuils, la valeur de la fonction objective (Otsu) et le temps d'exécution.

Les valeurs des seuils seront comparés avec ceux obtenus par une recherche exhaustive [23 (2010) 676–688, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem] et qui correspondent aux véritables valeurs optimales et qui sont consignés dans le tableau (3.2).

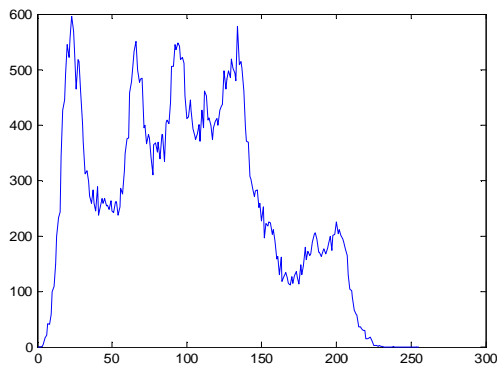


a: Image "Lena"

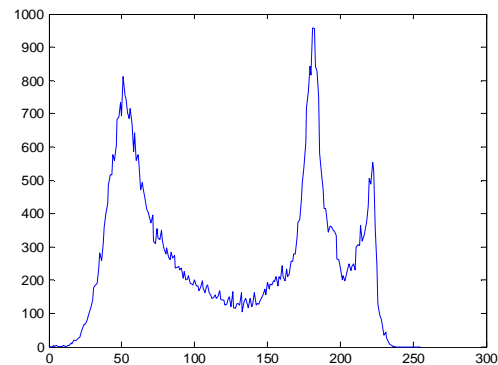


b: Image "Lac"

Figure (3.1) : Images Tests



a : Image "Lena"



b : Image "Lac"

Figure (3.2) : Histogrammes des images tests

Images	Nombre de seuils k	Les valeurs de seuils optimaux
Lena	1	102
	2	78-145
	3	57-106-159
	4	47-84-119-164
Lac	1	125
	2	88-155
	3	80-140-193
	4	69-111-158-197

Tableau (3.2) : Valeur des seuils obtenus par une recherche exhaustive

3.3.1 Résultats expérimentaux

Les tableaux (3.3) à (3.8) regroupent les résultats du seuillage à base de L'AED selon les différents modèles probabilistes.

Images	Nombre De seuils K	Les valeurs De seuils	Fonction objective d'Otsu	Temps En sec
Lena	1	102	1.1815e+004	0.444503
	2	78-145	1.2277e+004	0.479730
	3	57-106-159	1.2476e+004	0.506127
	4	48-86-118-167	1.2551e+004	0.595356
Lac	1	125	1.9220e+004	0.421918
	2	88-155	1.9495e+004	0.466084
	3	80-140-193	1.9631e+004	0.530280
	4	69-110-160-199	1.9697e+004	0.576869

Tableau (3.3) : Résultats du seuillage à base de L'AED et selon la loi normale

Images	Nombre De seuils K	Les valeurs De seuils	Fonction objective d'Otsu	Temps En sec
Lena	1	102	1.1815e+004	0.853903
	2	76-144	1.2276e+004	0.669477
	3	59-104-156	1.2474e+004	0.900969
	4	44-81-112-160	1.2547e+004	0.837729
Lac	1	125	1.9220e+004	0.618243
	2	88-157	1.9495e+004	0.645976
	3	83-147-194	1.9627e+004	0.823894
	4	71-103-160-196	1.9689e+004	0.993186

Tableau (3.4) : Résultats du seuillage à base de L'AED et selon la loi gamma

Images	Nombre De seuils K	Les valeurs De seuils	Fonction objective d'Otsu	Temps En sec
Lena	1	102	1.1815e+004	0.467042
	2	78-145	1.2277e+004	0.529895
	3	57-106-159	1.2476e+004	0.548585
	4	48-86-121-166	1.2553e+004	0.602164
Lac	1	125	1.9220e+004	0.435825
	2	88-155	1.9495e+004	0.494985
	3	79-139-192	1.9630e+004	0.557618
	4	68-103-153-193	1.9695e+004	0.650605

Tableau (3.5) : Résultats du seuillage à base de L'AED et selon la loi log- normale

Images	Nombre De seuils K	Les valeurs De seuils	Fonction objective d'Otsu	Temps En sec
Lena	1	102	1.1815e+004	0.463197
	2	83-147	1.2273e+004	0.482500
	3	55-104-154	1.2474e+004	0.556367
	4	47-80-111 -144	1.2517e+004	0.576286
Lac	1	126	1.9220e+004	0.427096
	2	96-160	1.9491e+004	0.535199
	3	70-135-200	1.9615e+004	0.579339
	4	69-89-141-189	1.9675e+004	0.614349

Tableau (3.6) : Résultats du seuillage à base de L'AED et selon la loi exponentielle

Images	Nombre De seuils K	Les valeurs De seuils	Fonction objective d'Otsu	Temps En sec
Lena	1	102	1.1815e+004	0.572171
	2	78-145	1.2277e+004	0.661415
	3	57-106-159	1.2476e+004	0.716186
	4	49-81-118-166	1.2551e+004	0.816585
Lac	1	125	1.9220e+004	0.625508
	2	89-156	1.9495e+004	0.717219
	3	81-139-187	1.9626e+004	0.738840
	4	67-105-160-198	1.9694e+004	0.868594

Tableau (3.7) : Résultats du seuillage à base de L'AED et selon la loi de Weibull

Images	Nombre De seuils K	Les valeurs De seuils	Fonction objective d'Otsu	Temps En sec
Lena	1	102	1.1815e+004	0.407942
	2	76-146	1.2276e+004	0.476663
	3	62-100-156	1.2464e+004	0.527695
	4	52-91-122-160	1.2547e+004	0.571132
Lac	1	125	1.9220e+004	0.418221
	2	89-158	1.9495e+004	0.524826
	3	81-141-183	1.9615e+004	0.536673
	4	64-114-163-204	1.9690e+004	0.608146

Tableau (3.8) : Résultats du seuillage à base de L'AED et selon la loi de Rayleigh

Les figures (3.3) et (3.4) montrent les images segmentées obtenues par L'AED avec la loi Normale comme modèle probabiliste



(a)



(b)



(c)

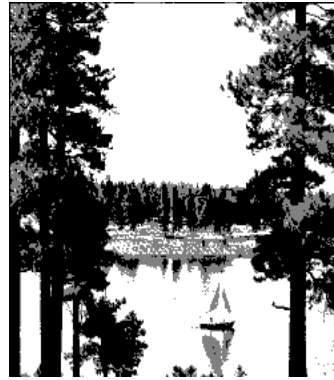


(d)

Figure (3.3) : Images Lena segmentées en utilisant la Loi Normale
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils



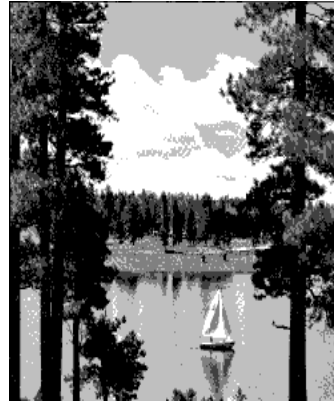
(a)



(b)



(c)



(d)

Figure (3.4) : Images Lac segmentées en utilisant la Loi Normale
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils

Les figures (3.5) et (3.6) montrent les images segmentées obtenues par L'AED avec la loi Gamma comme modèle probabiliste



(a)



(b)



(c)



(d)

Figure (3.5) : Images Lena segmentées en utilisant la Loi Gamma
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils



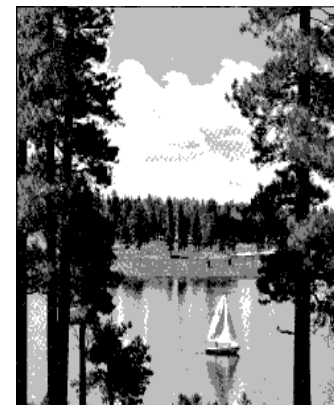
(a)



(b)



(c)



(d)

Figure (3.6) : Images Lac segmentées en utilisant la Loi Gamma
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils

Les figures (3.7) et (3.8) montrent les images segmentées obtenus par L'AED avec la Loi Log- normale comme modèle probabiliste



(a)



(b)



(c)



(d)

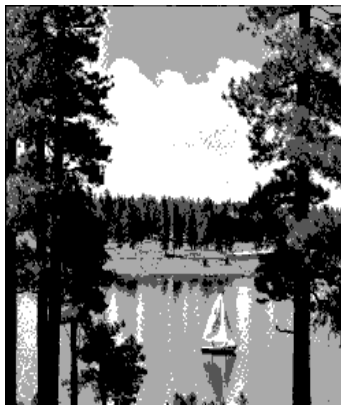
Figure (3.7) : Images Lena segmentées en utilisant la Loi Log-Normale
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils



(a)



(b)



(c)



(d)

Figure (3.8) : Images Lac segmentées en utilisant la Loi Log-Normale
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils

Les figures (3.9) et (3.10) montrent les images segmentées obtenues par L'AED avec la Loi exponentielle comme modèle probabiliste



(a)



(b)



(c)



(d)

Figure (3.9) : Images Lena segmentées en utilisant la Loi exponentielle
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils



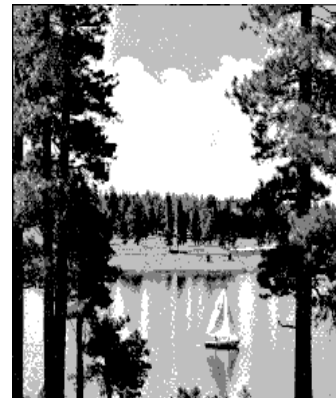
(a)



(b)



(c)



(d)

Figure (3.10) : Images Lac segmentées en utilisant la Loi exponentielle
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils

Les figures (3.11) et (3.12) montrent les images segmentées obtenues par L'AED avec la Loi de Weibull comme modèle probabiliste



(a)



(b)



(c)



(d)

Figure (3.11) : Images Lena segmentées en utilisant la Loi Weibull
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils



(a)



(b)



(c)



(d)

Figure (3.12) : Images Lac segmentées en utilisant la Loi Weibull
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils

Les figures (3.13) et (3.14) montrent les images segmentées obtenues par L'AED avec la Loi Rayleigh comme modèle probabiliste



(a)



(b)



(c)



(d)

Figure (3.13) : Images Lena segmentées en utilisant la Loi Rayleigh
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils



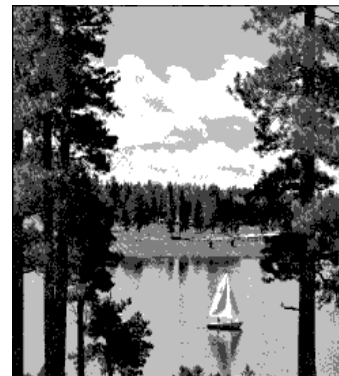
(a)



(b)



(c)



(d)

Figure (3.14) : Images Lac segmentées en utilisant la Loi Rayleigh
(a) : 1 seuil, (b) : 2 seuils, (c) : 3 seuils, (d) : 4 seuils

3.3.2 Interprétation des résultats

On remarque que toutes les solutions obtenues à base de l'algorithme à estimation des différentes lois de distribution (normale, log- normale, gamma,...) approchent les solutions optimales obtenues par une recherche exhaustive.

Lorsque le nombre de seuils égale à 1,2 ou 3, l'algorithme utilisant la loi Normale donne des solutions optimales.

Lorsque le nombre de seuils égale à 4, l'algorithme utilisant la loi Normale donne une solution plus approchée de la solution optimale.

En se basant sur le critère de comparaison qui est le temps d'exécution, la solution obtenue par l'algorithme utilisant la loi de Rayleigh comme loi de distribution approche la solution optimale en un temps d'exécution minimal et ce dans le cas de l'image Lena. Par contre dans le cas de l'image Lac c'est l'algorithme utilisant la loi Normale qui donne la solution optimale en un temps d'exécution minimal.

En se basant sur le critère de comparaison qui est le temps d'exécution et la qualité des solutions obtenues, l'algorithme utilisant la loi Normale comme loi de distribution est le plus performant.

3.4 Conclusion

Nous avons montré dans ce chapitre comment appliquer L'AED pour résoudre le problème du seuillage d'histogrammes en vue de la segmentation des images.

L'AED proposé pour résoudre le problème du seuillage est conçu sur le modèle UMDA (Univariate Marginal Distribution Algorithm) qui nécessite le choix d'un modèle probabiliste pour chaque distribution marginale.

Plusieurs modèles probabilistes décrits par des lois statistiques différentes ont été testés et comparés.

Les résultats obtenus montrent que le modèle probabiliste décrit par la loi Normale est le plus performant pour résoudre le problème du seuillage d'histogrammes.

Conclusion générale

Conclusion générale

Nous avons abordé dans ce mémoire un problème de segmentation d'images par seuillage d'histogramme en utilisant un Algorithme à estimation de distribution.

Le problème du seuillage consiste à calculer la valeur d'un ou plusieurs seuils en optimisant un critère donné. La recherche d'un seul seuil peut se faire facilement et rapidement d'une manière exhaustive. Cependant cette recherche exhaustive devient prohibitive lorsque le nombre de seuils augmente.

Nous avons ainsi proposé dans ce mémoire de rechercher rapidement les seuils en utilisant une méthode d'optimisation métaheuristique à base d'un algorithme à estimation de distribution.

L'algorithme à estimation de distribution est une classe des algorithmes évolutionnaires, il est utilisé pour résoudre des problèmes d'optimisation. Dans les méthodes de type AED, il n'y a pas d'opérateurs de croisement ou de mutation. En effet, la population de nouveaux individus est tirée au hasard, selon une distribution estimée depuis des informations issues de la population précédente.

L'AED proposé pour résoudre le problème du seuillage est conçu sur le modèle UMDA (Univariate Marginal Distribution Algorithm) qui nécessite le choix d'un modèle probabiliste pour chaque distribution marginale.

Plusieurs modèles probabilistes décrits par des lois statistiques différentes ont été testés et comparés.

Les résultats obtenus montrent que L'AED fournit de bons résultats et que le modèle probabiliste décrit par la loi Normale est le plus performant pour résoudre le problème du seuillage d'histogrammes.

Comme perspectives, nous préconisons d'utiliser le modèle multivariées afin de prendre en considération la dépendance entre les seuils. Comme il est intéressant de comparer les résultats de L'AED avec ceux obtenus avec d'autres algorithmes d'optimisation métaheuristique tels que : algorithmes génétiques, à essaim de particules...etc.

Bibliographies

- [1] : J. Dréo, A. Pétrowski, P. Siarry, E. Taillard : Les méta-heuristiques pour l'optimisation difficile .
- [2] : Imane Sebari et Dong- Chen HE : Les approches de segmentation d'image par coopération régions- contours, Revue télédétection, 2007, Vol.7.
- [3] : Sidd harta K. Shahya : Probabilistic model building Genetic Algorithm (PMBGA) : A survey, Technical Report, 2003.
- [4]: K. Hammouche, M. Diaf, P. Siarry : A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, 23 (2010) 676–688, 2010.
- [5]: K.Larbi: Segmentation d'images basée sur la modélisation statistique d'histogrammes, 2012.
- [6] : Y.Cooren, M.Clerc et P.Siarry : Optimisation par essaim particulaire améliorée par hybridation avec un algorithme à estimation de distribution, 2008, Vol.5.
- [7] : R. Armañanzas, I.Inza, R.Santana, Y. Saeys, J.L. Flores, J.A. Lozano, Y.V.D. Peer, R.Blanco, V.Robles, C. Bielza and P. Larrañaga : A review of estimation of distribution algorithms in bioinformatics, 2008.
- [8]: T.Miquélez, E.Bengoetxea, P.Larrañaga: Evolutionary computation based on Bayesian classifiers, Vol.14, 2004.
- [9]: R.Santana: An introductory course to Estimation of Distribution Algorithms.
- [10]: T.Lelore: Segmentation d'images Application aux documents anciens, 2007.
- [11] : I. Bloch, Y. Gousseau, H. Maître, D. Matignon, B. Pesquet-Popescu, F. Schmitt, M. Sigelle, F. Tupin : Le traitement des images, tome2, 2005.
- [12] : Métaheuristiques : un panorama.
- [13] : A.Nakib : Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application à des images biomédicales, 2007.
- [14]:T.K.Paul and H.Iba : Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms, 2002.
- [15]: L. Djerou, H.Dehipi, N.Khelil et M. Batouche : Self-Adaptive Thresholding for Image Segmentation.
- [16]: P-S.Liao, T-S.Chen and P-C.Chung: A Fast Algorithm for Multilevel Thresholding, 2001.
- [17]: P.Posik: Gaussian EDA and Truncation Selection: Setting Limits for Sustainable Progress.

- [18]: J-L.Baril: Cours de Traitement de l'Image.
- [19] : A.Manzanera : Traitement d'images et vision artificielle, Cours ENSTA D9-2.
- [20] : V.Bombardier : Traitement numérique de l'image (Numérisation- Mémorisation), 2008.
- [21] : J-P Lecoutre : Statistique et probabilités, Dunod, 4^{ème} Edition, 2009 (pp. 88-92).
- [22] :http://fr.wikipedia.org/wiki/Loi_normale
- [23] : http://fr.wikipedia.org/wiki/Loi_log-normale
- [24] : http://fr.wikipedia.org/wiki/Loi_Gamma
- [25] : http://fr.wikipedia.org/wiki/Loi_beta
- [26] : J.Dréo, P.Siarry. Métaheuristiques d'optimisation vues sous l'angle de l'échantillonnage de distribution.
- [27]: S.Baluja, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Pittsburgh, PA: Carnegie Mellon University (Technical Report No. CMU-CS-94-163), 1994.
- [28]: H.Mühlenbein & G.Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. Parallel Problem Solving from Nature, eds. Voigt, H.-M and Ebeling, W. and Rechenberg, I. and Schwefel, H.-P., LNCS 1141, Springer:Berlin, (pp. 178-187), 1996.
- [29]: D.E.Goldberg, Genetic Algorithms in search, optimization and machine learning. Addison-Wesley.0-201-15767-5, 1989.
- [30]: G.R.Harik, F.G.Lobo, & D.E.Goldberg. The compact genetic algorithm. In Proceedings of the IEEE Conference on Evolutionary Computation 1998 (ICEG'98) (pp. 523-528). Piscataway, NJ: IEEE Service Centre, 1998.
- [31]: J.S.De Bonet, C.L.Isbell, & P.Viola. MIMC: Finding optima by estimating probability densities. In M.C.Mozer, M.I.Jordan, & T.Petsche (editors), Advances in Neural Information Processing Systems, Volume 9 (pp. 424). The MIT Press, Cambridge, 1997.
- [32]: S.Baluja, & S.Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In Proceedings of the 14th International Conference on Machine Learning (pp. 30--38). Morgan Kaufmann, 1997.
- [33]: M.Pelikan, & H.Mühlenbein. The bivariate marginal distribution algorithm. In R.Roy, T.Furuhashi, & P.K.Chawdhry(Eds.), Advances in Soft Computing - Engineering Design and Manufacturing (pp. 521--535). London: Springer-Verlag, 1999.
- [34]: G.Harik, Linkage learning via probabilistic modeling in the ECGA. Urbana, IL: University of Illinois Genetic Algorithms Laboratory (IlligAL Report No. 99010), 1999.

- [35]: H.Mühlenbein, & T.Mahnig. Convergence theory and applications of the factorized distribution algorithm. *Journal of Computing and Information Technology*, Volume 7 (pp. 19-32). 1999a.
- [36]: H.Mühlenbein, & T.Mahnig. FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation* 7(4), (pp.353-376). 1999b.
- [37]: H.Mühlenbein, & T.Mahnig. *Evolutionary Algorithms and the Boltzmann Distribution. Foundations of Genetic Algorithms (FOGA2002)*. 2002.
- [38]: M.Pelikan, D.E.Goldberg, & E.Cantú-Paz, (1999). BOA: The Bayesian optimization algorithm. In W.Banzhaf, J.Daida, A.E.Eiben, M.H.GarzoN, V.Honavar, M.Jakiela, & R.E. Smith(editors). In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Volume I* (pp. 525-532). Orlando, FL: Morgan Kaufmann Publishers, San Francisco, CA.
- [39]: R.Santana, E.Ponce de León , A.Ochoa : The edge incident model. *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)* 1999:352-359.
- [40]: S.Shakya, J.McCall: Optimization by estimation of distribution with DEUM framework based on Markov random fields. *International Journal of Automation and Computing* 2007, 4(3):262-272.
- [41]: J.A.Gómez, J.L.Mateo , J.M.Puerta : EDNA: Estimation of dependency networks algorithm. *Bio-inspired Modeling of Cognitive Tasks, Second International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC, of Lecture Notes in Computer Science* 2007, 4527:427-436.
- [42]: R.Etxeberria, P.Larrañaga: Global optimization using Bayesian networks. *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)* 1999:151-173.
- [43]: J.Bernsen. Dynamic thresholding of grey-level images proc. *Internat. Conf. Pattern Recognition*.pp.1251-1255, 1986.
- [44]: A.P.Dempster, N.M.Laird and D.B.Rubin. Maximum likelihood from incomplete data via EM algorithm. *Journal of the royal statistical society*, vol.39, pp. 1-38, 1977.
- [45]: K.Takahashi, H.Nakatani and K.Abe. Color image segmentation using ISODATA clustering method. 2nd Asian conf. On computer Vision, vol.1, pp523-527, 1995.