

LES ALGORITHMES GENETIQUES APPLICATION A LA SEGMENTATION DES IMAGES

*LASSOUAOUI Nadia *, HAMAMI Latifa **, NOUALI Nadia **

** Division Informatique mobile*

Centre de Recherche sur l'Information Scientifique et Technique

{nlassouaoui, nanouali}@mail.cerist.dz

*** Ecole Nationale Polytechnique, Laboratoire Signal & Communications
10 Avenue Hacen Badi, BP 182, El Harrach, Alger.*

lhamami@mail.cerist.dz

1. Introduction

Les algorithmes génétiques sont des algorithmes d'exploration fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des structures les mieux adaptées, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède certaines des caractéristiques de l'exploration humaine. Ils sont une transcription directe du processus naturel d'évolution au monde de l'informatique.

Les possibilités espérées des mécanismes d'évolution ont conduit quelques chercheurs des années 1950 à vouloir les simuler pour les appliquer à l'ingénierie. Mais ces travaux n'ont pas été probants en raison des connaissances insuffisantes, à l'époque, de la génétique naturelle et aussi en raison des faibles performances des calculateurs alors disponibles. Mais depuis le début des années 60, la quantité de travaux sur les algorithmes évolutionnaires suit une croissance exponentielle (Goldberg, 1989).

En effet, les AGs ont des propriétés qui en font des candidats de choix pour résoudre des problèmes d'optimisation dans lesquels la taille de l'espace de recherche est importante, où les paramètres interagissent de manière complexe et où très peu d'informations sur la fonction à optimiser sont disponibles. Ces propriétés ont motivé leur application en vision et traitement d'images.

Après avoir présenté la théorie des AGs dans la section suivante, nous présentons les travaux effectués en vision et plus particulièrement en segmentation d'images, ensuite,

nous présentons en détails l'application d'un algorithme génétique à la segmentation des images et les améliorations apportées pour l'adapter au problème de la séparation des cellules biologiques du col utérin.

2. Fondements des algorithmes génétiques

Les AGs sont fondés sur :

- Une représentation chromosomique des solutions du problème.
- Une méthode pour générer une population initiale de solutions.
- Une méthode d'évaluation qui classe les solutions selon leurs aptitudes.
- Des opérateurs génétiques, qui définissent la manière dont les caractéristiques des parents sont transmises aux descendants.

Les AGs se caractérisent selon quatre axes qui sont :

- Ils utilisent un codage des paramètres, et non les paramètres eux-mêmes.
- Ils travaillent sur une population de points, au lieu d'un point unique.
- Ils n'utilisent que les valeurs de la fonction étudiée ; pas sa dérivée ou une connaissance auxiliaire.
- Ils utilisent des règles de transition probabilistes et non déterministes.

Il est donc clair qu'un AG manipule un ensemble d'individus à la fois, où chaque individu est caractérisé par une empreinte génétique, c.-à-d., un ensemble de chromosomes.

L'évolution des AGs se fait par deux mécanismes : la sélection naturelle et la reproduction ; la sélection naturelle fait que seuls les individus les mieux adaptés survivent et se reproduisent ; quant à la reproduction sexuée, elle assure un brassage de l'ensemble des solutions. Ce phénomène (sélection/reproduction) réitéré plusieurs fois sur une population produit après plusieurs générations des individus très bien adaptés au milieu dans lequel ils vivent.

2.1. Les étapes d'un AG

A chaque génération, l'AG standard décrit originellement par J.Holland (Goldberg, 1989 ; Holland, 1992 ; Man *et al.*, 1996 ; Schmitt, 2001 ; Petrowski) accomplit une génération lorsque les opérateurs génétiques ont été appliqués simultanément aux individus de la population, en effet, à chaque génération, un nouvel ensemble de créatures artificielles (des chaînes de caractères) est créé en utilisant des parties des meilleurs éléments de la génération précédente. Bien qu'utilisant le hasard, les AGs ne

sont pas purement aléatoires. Ils exploitent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance. Nous donnons les différentes étapes d'un AGS.

2.1.1. La représentation chromosomique

C'est un moyen de coder les solutions d'un problème ou encore les individus. Chaque solution est constituée d'une chaîne de symboles appartenant à un alphabet à priori de petite taille. Le symbole a un rôle dans les AGs similaire au rôle du gène dans la génétique naturelle. Le choix d'un code doit tenir compte de la complexité du processus de codage/décodage qui peut ralentir les calculs et influencer considérablement la convergence de l'AG.

Historiquement, le codage utilisé par les AGs était présenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'état. Ce type de codage a pour intérêt de permettre de créer des opérateurs de croisement et de mutation simples. Cependant, ce type de codage n'est pas toujours bon comme le montrent les deux exemples suivants :

- Deux éléments voisins en terme de distance de Hamming ne codent pas nécessairement deux éléments proches dans l'espace de recherche.
- Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du chromosome n'a pas alors forcément la structure du problème.

Les AGs utilisant des vecteurs réels évitent ces problèmes en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. La structure du problème est conservée dans le codage.

Il existe deux types de difficultés pour le choix du codage. D'une part celui-ci doit pouvoir être adapté au problème de façon à limiter au mieux la taille de l'espace de recherche, et aussi de façon que les nouveaux individus engendrés par les opérateurs génétiques (croisement, mutation, ...) soient souvent les plus significatifs ; c.-à-d., qu'ils puissent coder des solutions valides en respectant les contraintes du problème à résoudre. Notons que le choix d'un codage approprié est un élément critique dont dépend grandement l'efficacité d'un AG.

2.1.2. La nouvelle population

Le choix de la population initiale d'individus conditionne fortement la rapidité de convergence de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état en veillant à ce que les individus produits respectent les contraintes. Si des informations a priori sur le problème sont disponibles, il paraît bien évidemment naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence.

La nouvelle population est obtenue en remplaçant les plus mauvais individus de la population courante par leurs fils, reproduits par la phase précédente, si ses fils sont mieux qu'eux. La littérature des AGs (Goldberg, 1989 ; Holland, 1992 ; Man *et al.*, 1996 ; Schmitt, 2001 ; Petrowski) offre une collection fortement variée de stratégies dont il convient ici d'en citer quelques unes :

1. La nouvelle population est composée uniquement d'enfants. On laisse alors disparaître tous les individus de la population courante. L'inconvénient majeur de cette approche est le risque de perdre le meilleur individu.
2. Les enfants remplacent d'une façon régulière les individus les moins forts de la génération courante.
3. La nouvelle génération est constituée des n meilleurs individus de la population intermédiaire formée d'enfants et de parents.

2.1.3. La fonction d'évaluation

Elle prend en argument l'individu et lui associe un coût de performance. Cette fonction doit guider l'algorithme vers l'optimum en réalisant implicitement une pression de sélection dans cette direction. De plus, l'efficacité d'un AG s'appuie pour une grande part, sur la qualité de cette fonction. La mise au point d'une bonne fonction d'adaptation doit respecter plusieurs critères qui se rapportent à sa complexité et à la satisfaction des contraintes du problème. Lorsque la fonction d'adaptation apparaît excessivement complexe, consommant une importante puissance de calcul, la recherche d'une approximation plus simple est souhaitable.

2.1.4. Les opérateurs génétiques

Ces opérateurs sont la base des AGs, définissent la manière dont les individus se recombinent et s'agencent pendant la phase de reproduction. Les opérateurs qu'on retrouve le plus souvent sont : sélection, croisement et mutation.

2.1.4.1. Sélection

Cette opération est fondée sur le principe d'adaptation de chaque individu d'une population à son environnement, suivant la théorie de la sélection naturelle introduite par Charles Darwin. Ainsi, seuls les individus les plus aptes à répondre à certains critères seront sélectionnés à survivre et à se multiplier. La fonction d'évaluation de l'adéquation des individus évalue chaque chromosome de la population, et seuls ceux dont la qualité sera suffisante seront conservés d'une génération à une autre. D'autres individus prendront la place des plus mauvais.

N' individus seulement sur N ($N > N'$) sont remplacés dans la population pour obtenir la nouvelle génération, N : étant la taille de la population. Le nombre de remplacements ayants été fixé arbitrairement par l'utilisateur. Les $(N - N')$ individus qui survivent sont parmi les meilleurs. Il s'agit d'une stratégie dite élitiste. Les N' autres individus de la génération suivante sont obtenus suite à une sélection en fonction de la valeur d'adaptation de chaque individu et application des opérateurs de reproduction.

Dans ce qui suit, nous présentons deux procédures de sélection :

Soit une population de N individus, la sélection de chaque individu i dépend de sa probabilité Ps_i appelée probabilité de sélection. Dans le cas d'un problème de maximisation (recherche d'un maximum de la fonction d'adaptation), Ps_i est donné par la formule :

$$Ps_i = f_i / \left(\sum_{j=1}^N f_j \right)$$

[1]

où : f_i, f_j : sont respectivement les valeurs de la fonction d'évaluation pour les individus i et j .

Dans le cas de minimisation :

$$Ps_i = 1 - \left(f_i / \left(\sum_{j=1}^N f_j \right) \right)$$

[2]

Le principe de l'opérateur de sélection est donné par l'algorithme suivant :

Pour i=1 jusqu'à N faire

Générer aléatoirement un nombre réel R_i , tel que $R_i \in [0, 1]$.

Si $R_i < Ps_1$ alors

Sélectionner l'individu d'indice 1.

Sinon

Sélectionner l'individu d'indice k, $2 \leq k \leq N$

Tel que : $Ps_{k-1} < R_i \leq Ps_k$

Fsi

Fpour

La variance de cette méthode est élevée ; en effet, par malchance, il est possible qu'un individu de mauvaise qualité soit sélectionné pour la reproduction autant de fois qu'il y a d'individus à remplacer. Il est aussi possible qu'un individu ayant une bonne valeur d'adaptation ne soit jamais sélectionné. Ce phénomène est responsable de la dérive génétique qui permet à certains individus de « survivre » au détriment d'individus meilleurs. L'opérateur de mutation et la taille de la population suffisamment grande permettent de limiter ce risque.

Une des méthodes proposée pour résoudre ce problème est appelée méthode du rang. Elle consiste à ordonner les individus en fonction de leurs performances (meilleure adaptation) par ordre décroissant. Puis, on ordonne à chaque individu une probabilité de sélection relative à son rang au lieu de son évaluation. Ainsi, Ps_i de chaque individu i est :

$$Ps_i = (N - \text{rang}_i + 1) / \sum_{j=1}^N \text{rang}_j \quad [3]$$

Où : N est le nombre d'individus dans la population,

$\text{rang}_i, \text{rang}_j$: sont respectivement les rangs des individus d'indice i et j.

En fin de la sélection, on aura une nouvelle population appelée population intermédiaire (P_{int}) qui est constituée des individus ayant les bonnes valeurs d'adaptation qui sont sélectionnés plusieurs fois, et les mauvais vont disparaître de la population.

Pour créer les chromosomes de la nouvelle population, on fait appel aux mécanismes de croisement et de mutation.

2.1.4.2. Croisement

Le croisement consiste à combiner deux individus quelconques (dits parents) pour en ressortir deux autres individus (dits enfants) pas forcément meilleurs que les parents ; il existe plusieurs variantes de cet opérateur, mais en général il consiste à couper en un ou plusieurs points deux individus (aux mêmes endroits dans les deux individus) et à échanger les parties situées entre ces points, ces derniers sont générés aléatoirement.

Une autre variante est le croisement uniforme qui utilise une chaîne de bits générée aléatoirement et de même longueur que les individus appelée masque. Les gènes des individus initiaux sont échangés en fonction de cette chaîne aléatoire, lorsque le bit correspondant vaut 1. Le croisement uniforme de permutations a les avantages de la simplicité et selon son auteur (Syswerda, 1989), d'une bonne efficacité.

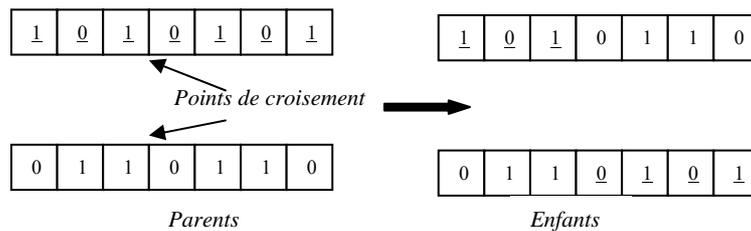


Figure 1 : Exemple du croisement à un point

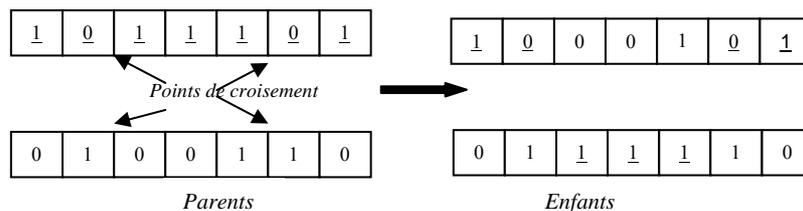


Figure 2 : Exemple du croisement à deux points

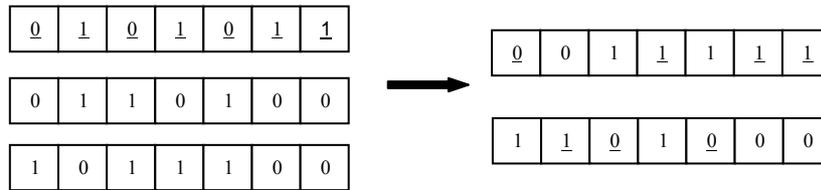


Figure 3 : Exemple du croisement uniforme

Il n'y a pas de critères de choix évidents entre les différentes variantes vues ci-dessus. Pour tel type de problèmes, le croisement « 1-point » sera décevant alors qu'un croisement uniforme fonctionnera mieux. Pour un autre type, la situation sera inverse. L'opérateur de croisement universel capable des meilleures performances pour tous les problèmes n'est pas encore découvert, et il est probable qu'il n'existe pas. Toutefois, en pratique, lorsque le codage binaire est satisfaisant, il apparaît que le choix de la bonne variante est rarement critique, et que les performances varient généralement peu selon l'opérateur choisi.

Un croisement est dit létal s'il produit un ou deux individus ayant une trop faible adaptation pour se reproduire. Une solution pour éviter une trop forte proportion de croisement des individus létaux consiste à accoupler préférentiellement les individus qui se ressemblent au niveau génotypique. Toutefois, lorsque le problème est complexe, bien qu'on ait l'assurance que les génotypes croisés ressemblent aux génotypes des parents, cela ne signifie pas que les descendants aient des performances proches de celles des parents. De plus, cette option réduit notablement l'exploration de l'espace de recherche et il peut s'ensuivre des convergences prématurées. Enfin, croiser des individus trop proches les uns des autres ne produit que rarement de la nouveauté, la convergence de l'algorithme s'en trouve ralentie. La restriction de l'appariement est donc une opération délicate à mettre en œuvre.

2.1.4.3. Mutation

Cet opérateur agit sur un individu. Il consiste à choisir d'une manière aléatoire un ou plusieurs gènes (bits) et à modifier leurs valeurs. Chaque bit d'un individu a une probabilité pour qu'il subisse une mutation, notée P_m . La mutation se fait comme suit :

Pour chaque bit de l'individu

Générer un réel aléatoire $r, r \in [0,1]$

Si $r < P_m$ alors le bit sera inversé.

Plusieurs arguments justifient l'utilisation de la mutation. Ainsi, les actions combinées de la pression de sélection et du croisement peuvent faire disparaître des gènes, aux mêmes positions des génotypes pour tous les individus. La mutation est capable de réintroduire ces gènes dans la population et participe ainsi au maintien de la diversité, utile à une bonne exploration de l'espace de recherche. Elle apporte aussi aux AGs la propriété d'ergodicité de parcours d'espace. Cette propriété indique que l'AG sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution.

D'autre part, elle assure une recherche locale aléatoire autour de chaque individu. Elle peut alors améliorer la qualité des solutions découvertes. En effet, le croisement perd de son importance lorsqu'une grande partie de la population est localisée dans les voisinages des maxima de la fonction d'adaptation. Dans ce cas, les individus situés sur un même pic sont souvent identiques et ne subissent aucune modification, ou alors, si les parents appartiennent à des pics différents, les descendants montreront généralement de faibles performances. En revanche, la recherche aléatoire due aux mutations donne une chance à chaque individu de s'approcher des positions exactes des maxima, autant que le permet la précision du codage.

2.1.5. Critère d'arrêt

Les étapes d'évaluation reproduction/sélection sont appliquées d'une manière itérative sur la population, jusqu'à satisfaction d'une condition Fin ; cette condition est soit un nombre maximum de générations, soit une valeur de fitness minimale, soit une convergence vers la meilleure solution.

Au cours de son évaluation, la population tend à converger, c.-à-d., que les individus tendent à se ressembler de plus en plus. Quand la population s'est uniformisée en grande partie, les individus fournissent une bonne approximation d'un optimum du problème. Si cet optimum n'est pas toujours l'optimum global, c'est généralement un optimum local proche (en qualité) de celui-ci.

2.2. Les variantes du modèle génétique

2.2.1. L'algorithme génétique canonique

Cet algorithme reste proche de l'algorithme génétique standard AGS sauf que la sélection se fait sur les descendants et leurs parents après la reproduction. La reproduction se fait sur des individus issus de la phase dite de clonage (figure 4).

Il s'agit d'une stratégie élitiste puisqu'elle conserve dans la population, d'une génération à l'autre, au moins l'individu ayant la meilleure adaptation. Il apparaît qu'une telle stratégie améliore considérablement les performances de l'AG pour certaines classes de fonctions, mais s'avère décevante pour d'autres classes, en augmentant les taux de convergence prématurés. En fait, une stratégie élitiste favorise l'exploitation des meilleures solutions, se traduisant par une recherche locale accentuée, au détriment de l'exploration de l'espace des génotypes.

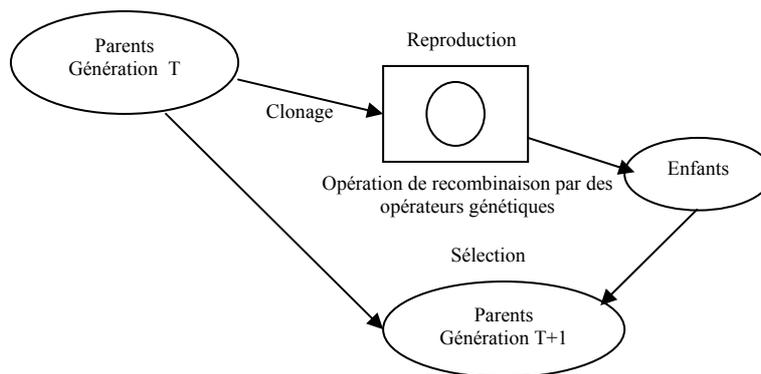


Figure 4 : Principe de l'algorithme canonique.

2.2.2. L'algorithme génétique modifié

L'algorithme génétique modifié AG-mod est un AGS modifié. La modification par rapport à l'AGS est que dans l'AG-mod, nous sélectionnerons indépendamment r chromosomes (pas nécessairement distincts) pour la reproduction et r chromosomes (distincts) pour mourir. Ces sélections sont faites selon l'aptitude relative des individus : un individu avec une meilleure performance a plus de chance d'être sélectionné pour la reproduction ; les individus avec des performances en dessous de la moyenne ont plus de chance à être sélectionnés pour mourir.

Une nouvelle population est formée, consistant en $N-r$ individus (tous les individus sauf ceux sélectionnés pour mourir) et r progénitures des r parents.

Evidemment, la sélection est exécutée de telle manière que les chromosomes les plus forts aient le moins de chance possible de mourir. Une des idées de l'AG-mod est une meilleure utilisation de la ressource disponible d'emménagement : la taille de la population.

A noter que pour toutes les variantes des AGs, les paramètres génétiques sont déjà fixés ; et deux exécutions successives de l'algorithme avec les mêmes paramètres peuvent converger vers des optima différents.

2.3. Les paramètres d'un AG

2.3.1. La taille de la population

Les conditions de convergence changent avec la taille de la population ; lorsque celle-ci est grande, sa diversité augmente ce qui diminue la convergence vers un optimum local, le temps d'exécution de chaque génération augmente, la recherche risque de s'effectuer d'une façon redondante et l'efficacité de l'algorithme est globalement affectée. Par contre, si la taille de la population est petite, alors la probabilité de s'attarder sur des minima locaux est grande. La taille selon le cas se situe entre 25 et 100.

2.3.2. Le taux de croisement

Il détermine la proportion des individus qui sont croisés parmi ceux qui remplaceront l'ancienne génération. L'opérateur de croisement est appliqué avec une probabilité P_c , et plus cette valeur est élevée plus de nouvelles structures (individus) sont introduites dans la nouvelle génération, les structures performantes sont trop fréquemment détruites. Par contre, si ce taux est trop bas, la population n'évolue pas assez vite. En général, P_c varie entre 0.25 et 0.70.

2.3.3. Le taux de mutation

L'opérateur de mutation est appliqué avec une probabilité P_m ; si ce taux est grand alors la recherche devient purement aléatoire, la population est diversifiée et l'AG perd de son efficacité. Si au contraire ce taux est faible, la population est moins diversifiée et en plus il y a risque de stagnation. Des études empiriques conseillent pour l'obtention de bons résultats une fréquence qui se situe autour d'une mutation tous les 1000 bits.

A noter que ces paramètres dépendent étroitement du type de problème à résoudre.

2.3.4. Le fossé des générations

K. DE JONG (Syswerda, 1989) a proposé de ne pas remplacer l'ensemble de la population à chaque génération et pour cela, il a introduit la notion d'écart entre les générations, un nombre compris entre 0 et 1 qui indique la proportion de parents qui

sont remplacés par leurs descendants. Si ce taux est à 1, l'ensemble de la population est remplacé.

2.4. Le parallélisme

L'intérêt de la parallélisation des AGs est de gagner en temps de calcul. Il existe pour cela au moins deux méthodes classiques utilisées :

La première consiste à diviser la population de taille N en n sous-populations et à les répartir sur l'ensemble des machines dont on dispose. L'intérêt est d'offrir la possibilité de travailler sur de grandes populations (N) tout en donnant des résultats dans un temps raisonnable puisque la durée nécessaire est à peu de chose près celle qu'il faudrait pour une population de taille N/n , si n est le nombre d'ordinateurs disponibles et si l'on néglige le temps de communication.

Notons que chaque machine a la possibilité de converger vers des optima qui seront différents de ceux calculés sur les autres ce qui correspond au comportement introduit avec le clustering. D'autre part, le surcoût de temps passé pour les communications n'est pas excessif puisqu'il n'y a de communication que de temps en temps. Il faut tout de même garder à l'esprit qu'une subdivision en sous-populations de taille trop réduite risque de conduire à faire tourner des AGs non fiables statistiquement. En effet, il faut quand même qu'une population contienne suffisamment d'individus pour que l'espace d'état puisse être exploré de façon correcte, afin que les résultats aient une certaine valeur.

La seconde maintient la population totale sur une seule machine mais les évaluations se font sur les autres machines afin qu'elles se fassent en même temps. Le principe se résume comme suit : un programme maître se charge de faire la sélection, les croisements, etc. En d'autres termes, il fait évoluer la population, puis répartit les calculs dont il a besoin sur un ensemble de démons. Enfin, dès qu'il reçoit tous les résultats, l'algorithme commence une nouvelle génération.

Il faut noter que ce mécanisme demande un grand nombre de communications pour envoyer les données et les évaluations. La méthode n'est donc intéressante que si le temps passé pour un calcul d'adaptation est grand devant le temps de communication, elle sera par conséquent utilisée pour des problèmes dont les évaluations de fonctions objectifs prennent du temps, on pense essentiellement à des cas faisant appel à des réseaux de neurones ou de gros calculs matriciels.

La parallélisation est extrêmement efficace pour accélérer les temps de résolution des AGs. Il faut certes bien étudier le problème afin d'utiliser le bon mécanisme de parallélisme.

Après avoir présenté la théorie des algorithmes génétiques, dans la section suivante, nous présentons les différents travaux effectués et plus particulièrement en vision.

3. Utilisation des algorithmes génétiques en analyse d'images

Ayant été reconnu comme une approche valide des problèmes nécessitant une exploration performante, et économique du point de vue calcul, les AGs sont maintenant appliqués plus largement aux domaines d'affaires, à la recherche scientifique, la finance, l'apprentissage, l'identification, la reconnaissance et les problèmes d'optimisation combinatoire dont un exemple typique : les "problèmes de permutation" (order based problems) ; qui consistent à découvrir un ordre des éléments d'une liste maximisant ou minimisant un critère donné (exemple : le problème du voyageur de commerce (Goldberg, 1989).

En analyse d'images, plusieurs travaux ont été présentés dans la littérature, dans ce qui suit, nous citons quelques uns.

3.1. Les algorithmes génétiques dans la segmentation des images

S. Bhandarkar et al (1994 ; 1999) considère le problème de la détection de contours comme un problème d'optimisation d'une fonction objective de coût en utilisant les AGs. Cette fonction initialement proposée par Tan et al (1991 ; 1989) utilise les informations sur les contours et les valeurs des niveaux de gris des régions.

En effet, Tan et al ont défini le problème de la détection de contours comme étant un problème de minimisation de coût, ils essayaient de surmonter les imperfections des techniques existantes (Gonzalez, 1987) en formulant une définition d'un contour qui est assez générale pour inclure la plupart des types de contours. Leur travail permet également d'améliorer les techniques existantes en considérant explicitement la structure locale du contour avec ses pixels voisins.

Par la minimisation de la fonction coût, les auteurs exigent que les contours détectés soient fins, continus, longs et d'une très bonne localisation. En effet, l'image de contours est obtenue par le calcul du coût qui est fonction de la dissemblance (ou dissimilitude) de la région, de la continuité, de la fragmentation et de la largeur des structures locales de contours. L'algorithme proposé est basé sur une technique simple de recherche

locale, il compare localement deux configurations de contours qui sont différentes uniquement en un point, et de sélectionner la meilleure.

Le coût ponctuel d'une image binaire de contours S à la position $p=(i,j)$ est obtenu par la somme pondérée des facteurs de coût C_i :

$$F(S, p) = \sum_i W_i C_i(S, p) \quad [4]$$

Les cinq termes de pénalité de la fonction coût sont :

C_d : la dissemblance entre les régions que séparent les contours,

C_t : l'épaisseur du contour,

C_c : la courbure du contour,

C_f : la fragmentation du contour,

C_e : le nombre de points de pixels détectés comme points contours.

Ces facteurs de coût sont calculés par recherche en arbre de décision (figure 5), ils sont destinés à favoriser la formation de contours entre régions de niveaux de gris moyens différents et à pénaliser les configurations dans lesquelles le nombre de points de contours est élevé, et à défavoriser les contours dont l'épaisseur, la courbure, ou encore la fragmentation sont fortes.

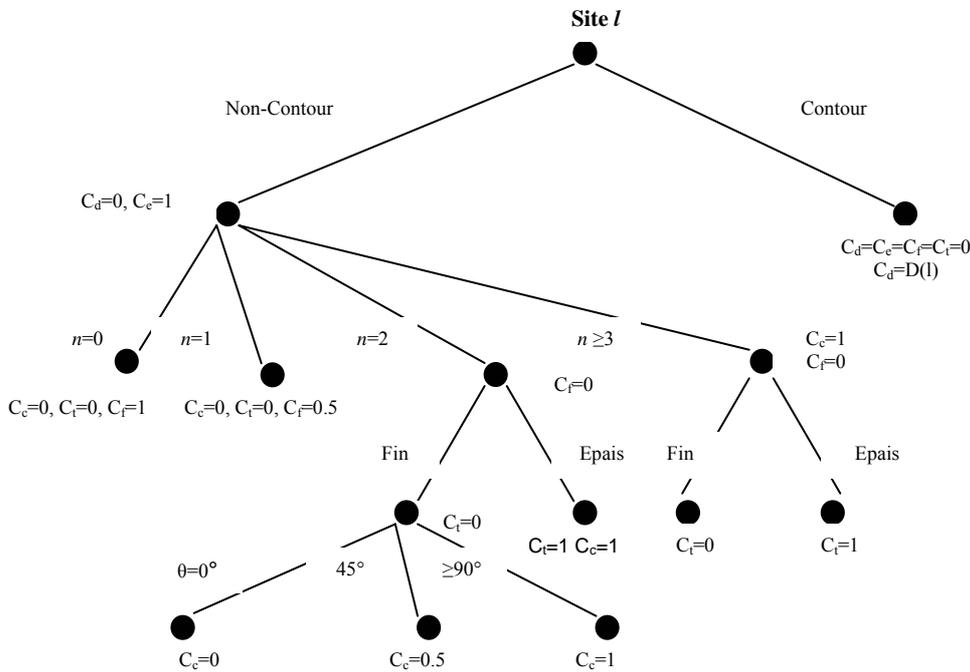


Figure 5 : Décision en arbre pour le calcul des facteurs de la fonction coût

Où : n correspond au nombre de pixels voisins, θ est l'angle associé à la structure du contour et $D(l)$ mesure la dissemblance entre régions séparées par le contour.

Bhandarkar et al (1994 ; 1999) ont proposé d'associer à cette approche les AGs pour la comparaison des structures de contours valides et tous les points contours (considérés sur un voisinage de 3×3) de l'image en utilisant une fonction fitness basée sur les facteurs de coûts.

Aussi, Bhanu et al (1991 ; 1995) ont construit un système de segmentation de scènes d'extérieur en couleur qui optimise, par AG, la segmentation de chacune des images qui lui sont présentées tout en gardant une mémoire de son expérience passée dans une base de règles du type 'condition/action'. Le système repose sur l'utilisation d'un AG de segmentation paramétrable et sur une fonction d'évaluation de la qualité des segmentations réalisées. La partie 'action' de chacune des règles code sous forme binaire, un jeu de paramètres de l'algorithme de segmentation. La partie 'condition' stocke les caractéristiques d'une image ainsi que l'heure et les conditions climatiques au moment de son acquisition. Lorsqu'une image est présentée au système, elle active les règles porteuses de parties 'condition' correspondant à cette image. Les parties 'action' de ces règles sont alors soumises à un AG dont l'objectif est de générer une paramétrisation conduisant à une segmentation de qualité satisfaisante, au sens de la fonction retenue a priori. Le nouveau jeu de paramètres ainsi obtenu est alors stocké, avec les caractéristiques de l'image, dans une règle de la base du système.

Dans l'approche développée par Chun et Yang (1996), se ne sont pas des configurations de contours qui évoluent mais des configurations de régions. Une partition initiale en un grand nombre de régions de taille éventuellement réduite est initialement obtenue par un algorithme de clustering. Un AG est ensuite utilisé pour améliorer cette segmentation en simulant des opérations itérées de fusion et de division des régions obtenues. Un chromosome porte autant de gènes qu'il y a de régions initiales et les allèles sont les labels affectés aux régions dans la segmentation pour laquelle il code. Un chromosome dans lequel des allèles des gènes i et j sont identiques spécifie une fusion des deux régions. Une mutation qui modifie l'un de ces deux allèles génère une séparation des deux régions.

3.2. Les algorithmes génétiques avec les réseaux de neurones pour la segmentation

Dokur et Olmez (2003) ont proposé un réseau de neurones quantifié QNN pour la segmentation des images à résonance magnétique MR et des images topographiques CT

(Computer Topography). Les éléments du vecteur caractéristique sont formés des intensités des pixels voisins du point considéré. QNN est une structure d'un réseau de neurones conçu par apprentissage avec les AGs. Chaque nœud de la première couche du QNN forme un hyperplan HP dans l'espace d'entrée. Il y a une contrainte sur les HPs dans les réseaux QNN. Le HP est représenté uniquement par un paramètre dans un espace d'entrée à N dimension. Les AGs sont utilisés pour trouver les valeurs optimales des paramètres des nœuds de ces réseaux QNN. Une comparaison pour la segmentation des images MR et des images CT en utilisant le perceptron multicouche et le réseau de kohonen a été donnée. Il a été observé que le QNN présente de meilleures performances de classification en utilisant peu de nœuds avec un temps d'apprentissage court.

Matsui et Kosugi (1999) proposent une méthode de segmentation des images par les réseaux de neurones classificateurs avec une sélection des paramètres basée sur les AGs. Ces derniers sont utilisés pour la sélection des combinaisons optimales des paramètres d'entrée du réseau de neurones classificateur. L'AG proposé est basé sur l'évaluation d'une fonction objective, qui permet d'estimer la combinaison optimale des paramètres des indices rapidement sans tester les classificateurs actuels. Les deux types de réseaux de neurones utilisés sont : le réseau de neurones par rétro propagation et le réseau de neurones à compteur propagation modifié (modified counter-propagation network).

3.3. Les algorithmes génétiques avec les contours actifs pour la segmentation

Une méthode de segmentation est proposée (Ballerini, 2001) en utilisant les contours actifs génétiques CAG (genetic snakes), qui sont un modèle actif de contours associé à une procédure de minimisation de l'énergie basée sur les AGs. Les CAG sont proposés afin de remédier à certaines limites des contours actifs classiques, telles que : l'initialisation, l'existence des multiples minima et la sélection des paramètres d'élasticité. Ballerini a prolongé la formulation des CAG de deux manières, en explorant le terme interne additionnel et le terme correspondant à l'énergie externe, des applications sur des images couleurs ont été données.

Aussi, Xu et al (2002) ont utilisé les AGs afin de trouver l'ensemble des paramètres optimaux d'un modèle de contours actifs (snakes MCA) pour la segmentation des lésions dans des images de mammographies. Ces paramètres sont typiquement estimés empiriquement et sont utilisés dans la fonction énergie qui est minimisée par l'algorithme MCA pour réaliser la détection des contours. En utilisant des détections de

contours manuelles fournies par des radiologues, les AGs sont utilisés pour faire varier les valeurs des paramètres jusqu'à ce que les contours produits par l'algorithme MCA soient très proches de ceux donnés par les radiologues.

3.4. Les algorithmes génétiques et les multifractals

Citons les travaux de Lassouaoui et al (2003) qui proposent un algorithme hybride multifractal-génétique pour la segmentation des images. L'algorithme s'exécute en deux étapes. La première effectue la recherche et la classification des pixels selon leurs valeurs en exposants de singularité (Lassouaoui *et al.*, 2002), suite à cette étape, nous obtenons des régions qui résultent du groupement des points ayant comme critère d'homogénéité une 'singularité proche'. Ensuite, une optimisation en utilisant les algorithmes génétiques est appliquée, l'objectif est de corriger les imperfections de la segmentation multifractale, tel que : combler les trous qui apparaissent dans les régions détectées ou éliminer/diminuer le chevauchement entre régions qui se touchent ou se collent. Une meilleure qualité de segmentation est obtenue.

3.5. Les algorithmes génétiques dans l'appariement stéréo

Dans (Jung *et al.*, 2000), un algorithme coopératif basé sur l'intégration de l'appariement stéréo et la segmentation. L'appariement stéréo est effectué à partir de deux images stéréo avec l'aide des résultats de la segmentation. En utilisant les AGs en segmentation, nous pouvons raffiner plus efficacement la carte en profondeur. Des résultats sont présentés pour illustrer les performances de la méthode proposée.

Les AGs sont aussi exploités dans le problème de fusion des résultats de segmentation, ou encore la reconnaissance et plusieurs autres applications, et la liste des travaux ne cesse d'augmenter.

Dans ce qui suit, nous présentons en détail un algorithme génétique pour la segmentation des images.

4. Application d'un Algorithme génétique à la segmentation des images

P. Andrey (1997) a proposé l'algorithme de relaxation sélectionniste qui est un algorithme génétique non supervisé de segmentation des images à plusieurs niveaux de gris qui nous fournit pour une même exécution l'image de contours et l'image de régions. Nous présentons en détails cet algorithme puisqu'il effectue la segmentation en

se basant uniquement sur un AG standard, ce qui montre l'exploitation des opérateurs génétiques à la segmentation.

4.1. Description de la méthode

L'adaptation d'une unité au contexte dans lequel elle se trouve plongée à la position qu'elle occupe sur l'image est l'un des aspects essentiels sur lequel repose le fonctionnement de cette technique.

Chaque unité est porteuse d'un chromosome qui la code pour une représentation putative de son environnement immédiat. Initialement quelconque, la répartition des unités se réorganise au fil des générations sous l'effet des opérateurs d'un AG standard distribué à grain fin. Une unité est un couple $U=(B, L)$. Le label L représente de manière symbolique l'appartenance de l'unité à une espèce donnée. Le chromosome B est un vecteur de caractéristiques qui constitue une représentation candidate de l'environnement de l'unité. Dans le cas de la segmentation au sens des niveaux de gris, il prendra des matrices de pixels, un codage réel des paramètres est adopté, cette représentation est plus appropriée qu'un codage binaire dans la mesure où les vecteurs de caractéristiques sont constitués de valeurs réelles du problème, et elle offre l'avantage d'éviter les barrières de Hamming.

Les unités peuvent être classées en deux catégories distinctes en fonction des unités qui les entourent. Les unités externes qui ont dans leur voisinage au moins une unité dont le label diffère du leur. A l'inverse, les unités internes se trouvent plus au centre de ces groupes et ne sont entourées que d'unités porteuses du même label que le leur. Les unités interagissent essentiellement avec les unités qui leur sont voisines, le voisinage retenu pour chaque unité est le huit voisins. A l'exception des unités sur les bords de l'image.

Dans ce qui suit, nous décrivons les différentes étapes de l'AG.

Initialisation : Les labels des unités sont initialement tous distincts. Nous partons donc d'une configuration dans laquelle chaque site est considéré comme une région distincte des autres. Les paramètres des vecteurs de caractéristiques des unités sont initialisés par tirage uniforme dans l'intervalle d'intensité de l'image, si elle est codée sur 8 bits, l'intervalle sera $[0,255]$.

Evaluation : La fitness $f_s(U)$ de l'unité U quantifie le degré de fidélité avec lequel son vecteur de caractéristiques représente les caractéristiques locales de l'image au

voisinage du site s qu'elle occupe. Elle doit être d'autant plus forte que l'écart entre les deux vecteurs est faible. La fonction fitness retenue est l'opposé de la distance de Manhattan entre le vecteur de caractéristiques de l'unité $B=(\beta_1, \dots, \beta_p)$ et le vecteur $A=(\alpha_1, \dots, \alpha_p)$ déterminé sur l'image :

$$f_s(U) = -\sum_{i=1}^p |\alpha_i - \beta_i| \quad [5]$$

s désignant le site occupé par l'unité. Notons que le label d'une unité n'intervient pas dans le calcul de sa fitness.

Sélection : La population d'unités est mise à jour de manière synchrone au cours de cette étape qui permet de sélectionner les individus les plus viables de la population précédente localement au voisinage de chaque site ou individu mais aussi globalement en explorant toute l'image à la recherche des meilleurs individus. Une sélection de type tournoi a été implantée, dont les modalités diffèrent suivant qu'elle est appliquée à un site d'une unité de type Externe ou de type Interne.

Si le site est occupé par une unité de type Interne : L'unité choisie pour occuper le site central et remplacer l'unité courante à la génération suivante est celle dont la fitness est la plus forte dans les unités au voisinage 3×3 de ce site. Si le site est occupé par une unité de type Externe : Une unité étrangère au voisinage est choisie au hasard et uniformément dans la population toute entière. Si sa fitness est meilleure que toutes les unités du voisinage du site, alors elle est choisie pour remplacer l'unité courante. Sinon, un tournoi local est effectué suivant les mêmes règles que celles appliquées au niveau d'un site occupé par une unité Interne.

Reproduction : Le croisement et la mutation entraînent une dynamique de l'étiquetage initial qui conduit à une configuration stable d'étiquettes et par conséquent à la formation des régions de l'image. Les unités de type Externe sont exclues de ces mécanismes. D'une part, pour éviter que les contours prennent un aspect très irrégulier à cause des effets éventuellement destructeurs de la recombinaison et de la mutation. D'autre part, cela empêche tout échange de matériel génétique entre unités porteuses de labels distincts puisqu'elles sont censées être adaptées à des caractéristiques, donc à des régions, distinctes de l'image. Elles pourraient aussi conduire à la formation de populations d'unités porteuses d'un même label bien qu'adaptées à des régions différentes, ce qui conduirait à une sous-segmentation.

Recombinaison : est l'uniforme généralisée. Pour chacune des unités Internes, une unité voisine quelconque est choisie comme partenaire de recombinaison. Notons que plus le taux de recombinaison est élevé, plus le coût moyen du chemin entre les segmentations initiale et finale est faible. Il est plausible que la recombinaison accélère la convergence en permettant l'apparition précoce de combinaisons d'allèles de bonne qualité, grâce aux échanges des produits des premières mutations. En l'absence de recombinaison, ces combinaisons apparaissent également, mais il leur faut plus de temps pour s'établir.

Mutation : est une mutation ponctuelle. L'amplitude de la perturbation subie par un allèle choisi pour être muté est proportionnelle à l'écart type σ_s des niveaux de gris dans le voisinage du site s calculée sur la fenêtre $k \times k$. Ainsi, la mutation du $i^{\text{ème}}$ allèle de l'unité U consiste à mettre à jour la valeur β_i selon :

$$\beta_i \leftarrow \beta_i + \varpi_s \quad [6]$$

où ϖ_s désigne un bruit gaussien centré de variance $m\sigma_s^2$. Où m contrôle l'amplitude de perturbation.

Notons que sur les N générations qui auront été effectuées au cours d'une relaxation, chacun des allèles de l'une des unités aura subi en moyenne $Pm.N$ mutations. Cependant, Les mutations successives étant indépendantes, leur effet cumulé pour un allèle donné sur l'ensemble des N générations conduit à une perturbation importante.

Cependant, dans le cas des images de scènes réelles, souvent l'estimation de σ_s n'est pas satisfaisante. En effet, souvent les régions dans la scène ne sont pas caractérisées par des variations locales et rapides des intensités lumineuses, mais sont au contraire assimilables à des surfaces qui résultent de l'illumination non uniforme de la scène. De telles régions ne satisfont pas le modèle stationnaire gaussien sur lequel repose la mutation et l'analyse de l'algorithme. Pour corriger l'effet de la variance, Andrey (1997) a proposé de multiplier la variance calculée sur une fenêtre $k \times k$ par un facteur correcteur.

Il faut noter que les valeurs du facteur correcteur proposées sont estimées empiriquement et que les performances de l'algorithme sont d'autant meilleures que la distribution des niveaux de gris des régions à segmenter est compatible avec le modèle

stationnaire gaussien, ce qui n'est pas toujours vérifié avec les images de scènes réelles.

Pour pallier à ces inconvénients, il est plus intéressant de choisir un mode de mutation qui s'adapte aux caractéristiques spatiales et d'intensité de chaque pixel. Ainsi, nous proposons d'effectuer la mutation comme suit :

« L'allèle du gène à la position i du chromosome à muter sera remplacé par l'allèle du gène à la position i du chromosome choisit aléatoirement parmi les 8 voisins ».

Ce choix nous a aussi permis d'éviter le calcul des variances au voisinage de chaque chromosome à muter, de rendre l'implémentation de la mutation plus simple et gagner en temps de calcul.

Aussi, pour diminuer l'effet de la mutation (Lassouaoui *et al.*, 2004) afin que la recherche ne devienne pas aléatoire au fur des générations, nous proposons d'appliquer la mutation avec une probabilité qui décroît selon la génération. En effet, des résultats de recherche (Lutton *et al.*, 1993 ; 1994) ont montré que des AGs à population finie et probabilité de croisement constante assurent théoriquement une convergence vers l'optimum global, si la probabilité de mutation P_m suit une loi de décroissance rapide en fonction de la génération k :

$$P_m(k) = P(0)\exp(-k/\alpha) \quad [7]$$

$P(0)$ est la probabilité de mutation initiale, α est calculé pour donner un taux final de mutation très bas (exemple de 10^{-4}) :

$$\alpha = (\text{Nombre maximale de générations}) / \ln(P(0)/10^{-4}) \quad [8]$$

Critère d'arrêt : Plusieurs critères peuvent être choisis, tel que : le temps d'exécution, la valeur fitness minimale, le nombre de générations, ...

Le nombre de générations N est choisie puisqu'il est nécessaire de l'adapter à la taille de l'image, dans la mesure où les unités se déplacent essentiellement de proche en proche. En effet, plus une image est grande, plus les régions qu'elle contient sont susceptibles de l'être également, et plus il faudra de temps aux unités pour les envahir.

Il est aussi nécessaire d'adapter la taille p des matrices de pixels utilisées comme vecteurs de caractéristiques au contenu de l'image à traiter. Il faut prendre plus ou moins suivant qu'on souhaite lisser ou non les détails fins de l'image.

4.2. Exemple d'application

Par la figure suivante, nous présentons un exemple d'application de la relaxation sélectionniste d'une image d'une cellule du col utérin, nous prenons les valeurs des paramètres de l'algorithme suivants : $p=3 \times 3$, $P_c=0.5$, $P(0)=0.03$, $N=150$, sachant que la taille de l'image est de 256×256 .

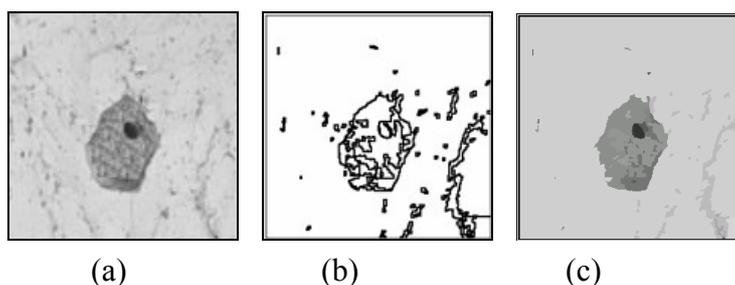


Figure 6 : Segmentation de l'image (a) par la relaxation sélectionniste.

Notons que la caractéristique principale de la relaxation sélectionniste est : « Le nombre de régions et leurs caractéristiques, informations sans lesquelles la segmentation ne peut être réalisée, sont déterminées automatiquement au cours du processus de relaxation ».

L'intérêt de la relaxation sélectionniste est sa capacité de détecter même les régions avec des intensités moyennes très proches de celles des régions voisines, c'est l'un des intérêts de cette technique spatialement adaptative, pour l'image 6(a), l'algorithme a détecté 16 régions (16 étiquettes). Notons que ces régions qui sont distinctes au sujet humain ne sont souvent pas détectées par la plupart des algorithmes de segmentation. Aussi, pour les régions où les variations locales des niveaux de gris sont beaucoup plus importantes, l'algorithme fournit une segmentation satisfaisante.

Sachant que l'algorithme travaille sur des données réelles, la non nécessité des étapes de codage et décodage rend l'algorithme rapide.

Afin d'adapter la segmentation obtenue à la reconnaissance des pathologies (Gompel, 1982) dans des images de cellules biologiques du col utérin, la segmentation doit permettre la détection des noyaux et cytoplasmes (Lassouaoui *et al.*, 2003), ainsi, nous proposons d'associer l'algorithme de Fisher (Cocquerez *et al.*, 1995) à la relaxation sélectionniste.

En effet, l'algorithme de multi-seuillage de Fisher permet de calculer les deux seuils (S_1 , S_2) qui définissent les trois principales classes : fond, noyaux et cytoplasmes. Ainsi, chaque région détectée par l'algorithme génétique sera classée par comparaison de sa moyenne d'intensité aux deux seuils (S_1 , S_2). Nous obtenons le résultat de la figure suivante. Un autre exemple est donné par la figure 8 où 28 régions sont détectées par la relaxation sélectionniste (figure 8.b).

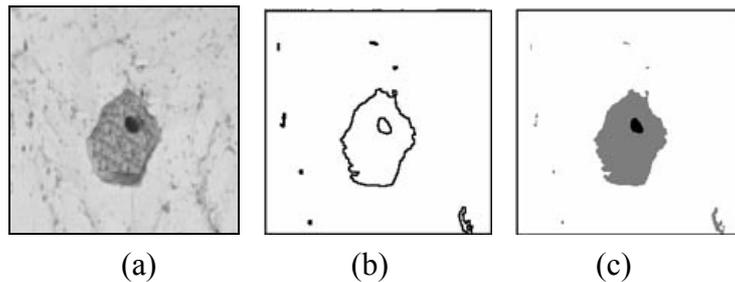


Figure 7 : Segmentation de l'image (a) par la relaxation sélectionniste et la classification par Fisher.

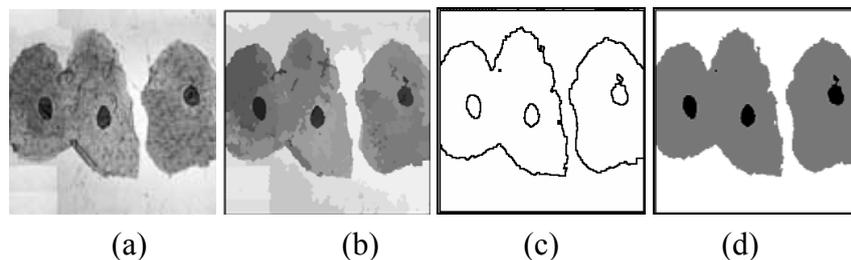


Figure 8 : Segmentation de l'image (a), (b) par la relaxation sélectionniste, (c) et (d) par la relaxation et la classification par Fisher.

Nous obtenons une meilleure qualité de segmentation, avec des résultats qui sont plus exploitables pour la reconnaissance puisque les régions segmentées correspondent aux réels objets de la scène ; et ceci en un temps d'exécution court, ce qui est important puisque les AGs sont généralement consommateurs en temps de calcul.

5. Conclusion

Les algorithmes génétiques sont des algorithmes d'exploration robustes et convergent vers une solution satisfaisante, lorsque leurs paramètres (taille de la population, nombre d'itérations, probabilités de tirages,...) sont choisis d'une manière adéquate. Cependant, le choix des paramètres ainsi que le codage des données est encore plus un art qu'une science. Il n'y a aucun jeu de paramètres qui serait universel pour tous les problèmes considérés du fait que ces valeurs dépendent étroitement du type de problème à résoudre. Un autre inconvénient est que pour le moment, on ne sait pas prédire qu'un AG convergera vers la bonne solution

Les AGs ont fait leurs preuves concernant la résolution de certains problèmes d'optimisation complexes. De grandes classes d'applications sont envisagées à ce jour par utilisation des AGs. Nous avons présenté différents travaux effectués dans des applications de la vision et de la segmentation.

Notons que les AGs ont été appliqués à la segmentation d'images, soit, pour optimiser les paramètres d'un algorithme de segmentation existant, soit, pour rechercher une configuration optimale dans l'espace des segmentations possibles. Cependant, ils se heurtent à certaines difficultés, en particulier :

Dans tous les cas, il est nécessaire de définir une fonction de fitness qui mesure la qualité de segmentation. Il n'existe pourtant pas de mesure communément admise, et des mesures distinctes peuvent conduire à des classifications distinctes. De plus, la fonction de fitness peut elle-même dépendre de paramètres qu'il faudra ajuster. Il est également nécessaire de déterminer les valeurs des paramètres qui régissent le comportement de ces algorithmes et l'état actuel de la théorie des AGs ne permet pas de disposer de règles automatiques pour fixer les valeurs de ses paramètres, et il est souvent nécessaire de procéder à un ajustement empirique.

Nous avons présenté et étudié en détail la relaxation sélectionniste qui dépasse les limitations déjà mentionnées (Andrey, 1997). Cette approche affine le résultat du

groupement des pixels après chaque génération issue de la population initiale par un ensemble de transformations génétiques de sélection, croisement et mutation.

Afin d'augmenter la qualité de la segmentation par la relaxation sélectionniste et aussi de l'adapter à des applications bien précises, telle que la séparation des cellules biologiques du col, nous avons proposé de lui associer l'algorithme de Fisher. Une meilleure qualité de segmentation est obtenue.

Les efforts futurs devront porter sur :

- La définition d'une stratégie qui permet d'effectuer la segmentation par adaptation des opérateurs, en sélectionnant de manière adéquate l'opérateur le plus approprié au type d'image.
- La définition d'un critère d'arrêt fondé sur la stabilité de la segmentation. Souvent les méthodes de segmentation génétiques consistent à effectuer un nombre de générations suffisamment élevé pour qu'on soit certain que la stabilité ait été atteinte. Cependant, le nombre de générations nécessaires à la convergence dépend en partie du contenu des images à traiter. Il est en outre soumis à la variabilité intrinsèque de la méthode résultant du caractère stochastique de la procédure d'initialisation des unités et des opérateurs génétiques utilisés. Il est donc nécessaire de disposer d'un autre critère que le nombre de générations pour arrêter le programme.

6. Références :

Abdulhafour M., Fellah A., « A hybrid approach of genetic algorithms and fuzzy logic applied to feature extraction from multisensory images », *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, 1997, p. 199-204.

Alimi A. M., « Evolutionary computation for the recognition of on-line cursive handwriting », *IEEE Journal of Research*, vol. 78, n°5, 2002, p. 385-396.

Andrey P., Tarroux P., « Unsupervised Segmentation Using a Distributed Genetic Algorithm », *Pattern Recognition*, vol. 27, n°5, 1994, p. 659-673.

Andrey P., Segmentation d'images par algorithmes génétiques, Thèse de doctorat, Université de Paris 7, 1997.

Andrey P., « Selectionist relaxation: genetic algorithm applied to image segmentation », *Image and vision computing*, vol. 17, 1999, p. 175-187.

Ballerini L., « Genetic snakes for color images segmentation », *Workshops on Applications of Evolutionary Computing*, vol° LNCS 2037, 2001, p. 268-77.

Bhandarkar S. M., Zhang Y., Potter W. D., « An edge detection technique using genetic algorithm based optimisation », *Pattern Recognition*, vol. 27, n° 9, 1994, p. 1159-1180.

Bhandarkar S. M., Zhang H., « Image segmentation using evolutionary computation », *IEEE Transactions on Evolutionary Computation*, vol. 3, n° 1, 1999, p. 1-21.

Bhanu B., Lee S., Ming J., « Self-Optimizing image segmentation system using a genetic algorithm », *Proceeding of the 4th International Congress on Genetic Algorithms*, 1991, p. 362-369.

Bhanu B., Lee S., Ming J., « Adaptive Image segmentation Using a Genetic Algorithm », *IEEE Transactions on Systems, Man, And Cybernetics*, vol. 25, n° 12, 1995, p. 1543-1180.

Bhanu B., Lee S., Das S., « Adaptive image segmentation using genetic and hybrid search methods », *IEEE transactions on Aerospace and Electronic Systems*, vol. 31, n° 4, 1995, p. 1268-1291.

Chun D. N., Yang H. S., « Robust image segmentation using genetic algorithm with a fuzzy measure », *Pattern-Recognition*, vol. 29, n° 7, 1996, p. 1195-1211.

Cocquerez J.P., Philipp S., *Analyse d'images : filtrage et segmentation*, Edition Masson, 1995.7

De Jong K., « Learning with Genetic Algorithms: An Overview », *Machine Learning*, vol. 3, 1988, p. 121-138.

Dokur Z., Olmez T., « Segmentation of MR and CT images by using a quantiser neural network, *Journal of Neural Computing Applications*, vol. 11, 2003, Springer-Verlag, p. 168-177.

Fan Y., Jiang T., Evans D. J., « Volumetric segmentation of brain images using parallel genetic algorithms », *IEEE-Transactions-on-Medical-Imaging*, vol. 21, n° 8, 2002, p. 904-909.

Goldberg D. E., *Algorithmes Génétiques : Exploration, optimisation et apprentissage automatique*, Edition Wesley, 1989.

Gompel C., *Atlas de la cytologie*, Ecole de Médecine de Paris, 1982.

Gonzalez R.C., *Digital Image Processing*, second edition, Addison Wesley, 1987.

Holland J. H., « Genetic Algorithms », *pour la science*, n°179, 1992, Edition of Scientific American, p. 44-50.

Jiang T., Fan Y., « Parallel genetic algorithm for 3D medical image analysis », *IEEE International Conference on Systems, Man and Cybernetics*, vol. 6, 2002, Tunisia.

Jung K., Han J., « Stereo correspondence using GA-based segmentation », *Second International Conference on Data-Mining, Financial Engineering and Intelligent Agents*, vol. LNCS 1983, 2000, p. 497-502.

Lassouaoui N., Hamami L., « Segmentation des Images de Cellules Biologiques par une Approche Multifractale », *9^{ème} Conférence Internationale sur le Traitement d'Information et Gestion d'Incertitudes dans les systèmes à Base de Connaissance IPMU' 2002*, vol. 3, juillet 2002, France, p. 2005-2011.

Lassouaoui N., Belouchrani A., Hamami L., « On the Use of Multi-fractal Analysis and Genetic Algorithms for the Segmentation of Cervical Cell Images », *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, n°7, 2003, World Scientific Publishing Company, p. 1227-1244.

Lassouaoui N., Hamami L., « Réalisation d'une base de données pour le module de reconnaissance du système de dépistage du cancer du col utérin », *Conférence Internationale de IEEE : Sciences Electroniques, Technologies de l'Information et des Télécommunications SETIT 2003*, 17 -21 Mars 2003, Sousse, Tunisie.

Lassouaoui N., Hamami L., « A New Mutation Operator for the Segmentation by Selectionist Relaxation of Real Images », *International Conference on Modelling and Simulation MS'2004*, 2004, Lyon, France, p. 9.5-9.8.

Lutton E., Martinez P., Détection de primitives géométriques bidimensionnelles dans les images à l'aide d'un algorithme génétique, Rapport de recherche n° 2110, novembre 1993, INRIA.

Lutton E., Martinez P., « A genetic algorithm for the detection of 2D Geometric Primitives in images », *Computer Vision and Image Processing*, vol. 1, 1994, p. 526-528.

Man K.F., Tang K.S., Kwong S., « Genetic Algorithms: Concepts and Applications », *IEEE Transactions on Industrial Electronics*, vol. 43, n°5, 1996, p. 519-534.

]

Maslov I.V., Gertner I., « Efficient image reduction for image registration with evolutionary algorithm », *SPIE International Society for Optical Engineering*, vol. 4787, 2002, Seattle, USA, p. 198-209.

Matsui K., Kosugi Y., « New image segmentation method by modified counter-propagation network and genetic algorithm », *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, 1999, Tokyo, Japan, p. 854-858.

Matsui K., Kosugi Y., « Image segmentation by neural-net classifiers with genetic selection of feature indices », *International Conference on Image Processing*, vol. 1, 1999, Japan, p. 524-528.

Petrowski A., Une introduction à l'optimisation par algorithmes génétiques, <http://www-inf.int-evry.fr/~ap/EC-tutoriel/Tutoriel.html>.

Ramos V., Muge F., « Map segmentation by colour cube genetic k-Mean clustering », *Fourth European Conference on Research and Advanced Technology for Digital Libraries*, vol. LNCS 1923, 18-20 Sept. 2000, Lisbon, Portugal , p. 319-230.

Schmitt L.M., « Fundamental study: Theory of genetic algorithms », *Theoretical Computer Science*, vol. 259, 2001, p. 1-61.

Syswerda G., « Uniform crossover in Genetic Algorithms », *3rd International Conference on Genetic Algorithms*, 1989, USA, p. 2-9.

Takagi K., Koike A., Matsumoto S., Yamamoto H., « Motion picture coding based on region segmentation using genetic algorithm », *Systems and Computers*, vol. 33, n°5, 2002, Japan, p. 41-50.

Tan H. L., Gelfand S. B., Delp E. J., « A comparative cost function approach to edge detection », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, n°6, 1989, p. 1337-1349.

Tan H. L., Gelfand S. B., Delp E. J., « A cost minimization approach to edge detection using simulated annealing », *Transactions on Pattern and Machine Intelligence*, vol. 14, 1991, p. 3-18.

Wada S., Imai N., « Efficient structural segmentation of scene image with genetic algorithm for merging boundary regions », *IASTED International Conference on Signal and Image Processing*, Aug 2001, Honolulu , p. 33-37.

Xu Y., Neu S.C., Ornes C.J., Owens J. F., Sklansky J., D.J. Valentino, « Optimization of active-contour model parameters using genetic algorithms: segmentation of breast lesions in mammograms », *Proceedings of the SPIE International Society for Optical Engineering*, 21-28 Feb. 2002, San Diego, p. 1406-1414.

Yen G., Nithianandan N., « Facial feature extraction using genetic algorithm », *Congress on Evolutionary Computation*, vol. 2, 2002, p. 1895-1900.

Zheng H., Kong L. X., Nahavandi S., « Automatic inspection of metallic surface defects using genetic algorithms », *9th International Conference on Metal Forming*, vol. 125-126, 2002, Birmingham, UK, p. 427-333.