

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique  
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE  
FACULTE DES MATHEMATIQUES



Métaheuristiques pour le Recalage des Images en 2D,  
cas des Images Satellitaires \*

Présenté par :

Lahlou BENGHEZAL \*\*

**Résumé :** Un des problèmes les plus importants lors du traitement et d'analyse d'images satellitaires dans tous les domaines que manipule la télédétection est le recalage qui conditionne fortement l'exactitude et la cohérence des interprétations des images satellitaires. Ce dernier, étant la maximisation de similarité entre images, devient un problème d'optimisation. Nous pouvons donc envisager n'importe quelle méthode d'optimisation pour résoudre ce problème.

Comme ces problèmes contiennent souvent beaucoup de solutions à intérêts pratiques acceptables, les spécialistes de l'optimisation combinatoire ont orienté leur recherche vers le développement des méthodes heuristiques. Le but est de trouver une solution de qualité satisfaisante en un temps de calcul raisonnable. D'autant plus que pour des problèmes réels, il n'est pas toujours impératif de trouver la solution optimale, mais des solutions dont la qualité et le temps pour l'obtenir restent dans l'acceptable. Ces performances étant de natures opposées, il s'agit alors de trouver un compromis selon le contexte du problème de recalage, et de valider l'étude théorique par des tests expérimentaux sur des images satellitaires.

Nous avons développé un système pour le **recalage dense** basé sur la maximisation de similarité entre images par deux métaheuristiques, le **recuit simulé** (en codage naturel et binaire) et les **algorithmes génétiques** (en codage naturel, binaire et quantique).

Pour réduire le temps d'exécution du processus de recalage qui peut être prohibitif quand la taille des images est grande, nous avons adopté des schémas d'accélération basés sur la minimisation de la quantité d'informations à traiter et la parallélisation des méthodes d'optimisation.

\* Thèse de Magister.

\*\* Directeur de thèse : Mr OUAFI Rachid Maître de conférence à l'USTHB.

---

# Introduction Générale

---

Un problème d'optimisation combinatoire est un problème qui peut s'exprimer par une fonction (dite de coût) avec ou sans contraintes, à minimiser ou maximiser sur un ensemble de définition fini ou dénombrable. C'est le cas de nombreux problèmes, dans des domaines d'applications très variés, qu'ils soient scientifiques ou techniques. Pour illustrer ce propos on peut citer des problèmes académiques tels que la coloration de graphes ou le sac à dos multidimensionnel, ainsi que des applications réelles comme le positionnement de composants dans la conception de circuits imprimés, la définition de réseaux de radio émetteurs ou la restauration des images. Aussi, l'optimisation combinatoire est un domaine qui fait l'objet de recherches intenses.

Nous avons choisi comme champ d'étude le recalage en imagerie satellitaire, plus précisément le recalage rigide et le recalage déformable (tout deux denses, i.e. basés sur les niveaux de gris, en 2D). Dans ce contexte, le but du recalage est la mise en correspondance des structures géographiques afin de suivre, par exemple, l'évolution de constructions ou d'évaluer l'ampleur d'un désastre. De nombreuses méthodes de recalage en imagerie satellitaire consistent à minimiser une fonction de coût, ou fonction de similarité exprimant la similitude au niveau des pixels en 2D des images que l'on cherche à recaler entre elles. Or les fonctions de similarité classiques en recalage des images sont non linéaires, irrégulières et présentent de nombreux minima locaux. Pour résoudre ce problème d'optimisation difficile, il est donc nécessaire de recourir à un algorithme d'optimisation globale.

## **Contribution de ce mémoire**

L'objet de ce Mémoire porte sur l'étude de deux méthodes heuristiques et une approche algorithmique pour résoudre le problème du recalage des images satellitaires. Le but final est de constituer une bibliothèque d'algorithmes à utiliser selon le choix de l'utilisateur (Choix des méthodes, critère d'optimalité, temps de calcul...). Des analyses des résultats et les comportements des différents algorithmes permettront de conclure sur l'efficacité de chaque algorithme en fonction des critères à optimiser.

Le problème particulier à étudier est le recalage des images 2D. Le recalage permet de faire, dans le cas des images satellitaires et/ou aériennes, la fusion des images et la mise en correspondance des zones géographiques afin de déceler par exemple l'évolution de l'environnement, l'évaluation des changements et l'analyse thématique (ex des nouvelles constructions, de la verdure ou la sécheresse et détecter les différents mouvements des engins). De nombreuses méthodes de recalage en imagerie consistent à minimiser une fonction de coût exprimant la similitude au niveau des pixels en 2D des images que l'on cherche à recaler entre elles.

Ces fonctions sont généralement non linéaires, irrégulières et présentent de nombreux minima locaux, d'où l'utilité du recours aux algorithmes d'optimisation globale.

L'étude concernera l'implémentation de plusieurs méthodes d'optimisation qui sont les algorithmes génétiques, le recuit simulé et pour pallier au coût élevé des calculs dans le cas de l'application du recalage, on se propose d'accélérer les méthodes d'optimisation.

# Chapitre I : Recalage des images satellitaires (état de l'art)

---

## Introduction

Le recalage (ou la mise en correspondance) des images est un problème d'optimisation car il se traduit par l'optimisation d'une certaine fonction, donc on peut utiliser des méthodes d'optimisation pour le résoudre, mais bien avant il faut bien étudier les caractéristiques (surtout la complexité) du recalage afin de bien choisir une méthode adéquate.

Pratiquement, le recalage des images est une tâche très précieuse, dans le sens où plusieurs applications de vision par ordinateur font l'intervenir. L'application de la mise en correspondance dans le cadre d'imagerie satellitaire, importe un bénéfice important à cause du nombre important des tâches de diagnostic et/ou thématique même de recherche complexe, qui devient très évidentes en cas d'utilisation du recalage. Citons par exemple la fusion des images satellitaires et/ou aériennes qui permet de localiser avec précision des éventuels changements, l'approche manuelle peut être fastidieuse (sans parler de la précision). Ce qui explique le nombre important des projets lancés et les articles publiés dans ce sens.

Toujours dans le cadre de l'imagerie satellitaire et comme on le verra plus tard, le recalage peut être utilisé pour une simple mise en même échelle ou une fusion de données qui ne nécessitent qu'une simple transformation avec peut de variables, à la création des atlas géographiques ou des cartographies opérationnelles qui nécessitent des transformations plus complexes dont le nombre de variables est plus important. Ce qui régit la variation de complexité des recalages, dans ce qui suit on va essayer donc à énumérer les paramètres qui contrôlent la complexité d'une méthode de recalage, et ensuite d'essayer de les classées.

Pour interagir, et surtout comprendre ce qui se passe autour de lui, l'être humain utilise ses cinq sens. D'après les psychos physiologistes, la vue est le sens primordial pour l'être humain, le système de vision humain derrière l'œil est, donc, le moyen de communication et d'interaction le plus efficace et le plus important. Cette importance n'est pas seulement due à la puissance et à la complexité du système de vision humain, mais aussi à la richesse des images en terme d'informations portées par elles.

La vision par ordinateur était parmi les premières voies proposées avec la révolution de l'informatique, en réalité, les ordinateurs deviennent de plus en plus puissants. Cette vision par ordinateur n'avait pas de sens sans une définition de l'image compatible avec les principes numériques qui caractérisent les informations manipulées par l'ordinateur. Cette définition ne pouvait se faire sans un processus de numérisation d'images, qui sera vue comme étant « la projection bidimensionnelle contenant des points caractérisés par l'intensité lumineuse perçue par un capteur quelconque », cette unité s'appelle le pixel, qui devient l'unité de base pour la construction d'images.

## Téledétection

L'information géographique est une denrée de plus en plus précieuse, que ce soit à des fins commerciales (exploitation des ressources terrestres), stratégiques (surveillance des territoires) ou humaines (aide au développement, prévention des catastrophes). Depuis un siècle on a su « prendre de la hauteur » pour observer la terre à bord de ballons, fusées, avions, hélicoptères, drones, satellites et navettes spatiales. La téledétection est l'ensemble des techniques qui permettent d'acquérir à distance des informations, en général des images, et de les traiter pour répondre à des problèmes.

La télédétection est la technique qui, par l'acquisition d'images, permet d'obtenir de l'information sur la surface de la terre sans contact direct avec celle-ci. La télédétection englobe tout le processus qui consiste à capter et à enregistrer l'énergie d'un rayonnement électromagnétique émis ou réfléchi, à traiter et à analyser l'information, pour ensuite mettre en application cette information.

Dans la plupart des cas, la télédétection implique une interaction entre l'énergie incidente et les cibles. Le processus de la télédétection au moyen de systèmes imageurs comporte les sept étapes que nous élaborons ci-après.

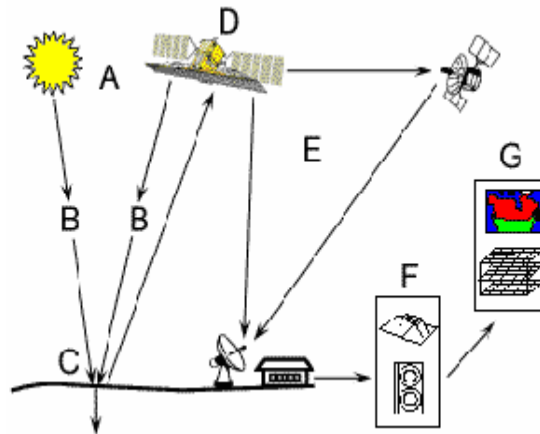


Figure 1. Les étapes de la télédétection.

1. **Source d'énergie ou d'illumination (A)** - À l'origine de tout processus de télédétection se trouve nécessairement une source d'énergie pour illuminer la cible.
2. **Rayonnement et atmosphère (B)** - Durant son parcours entre la source d'énergie et la cible, le rayonnement interagit avec l'atmosphère. Une seconde interaction se produit lors du trajet entre la cible et le capteur.
3. **Interaction avec la cible (C)** - Une fois parvenue à la cible, l'énergie interagit avec la surface de celle-ci. La nature de cette interaction dépend des caractéristiques du rayonnement et des propriétés de la surface.
4. **Enregistrement de l'énergie par le capteur (D)** - Une fois l'énergie diffusée ou émise par la cible, elle doit être captée à distance (par un capteur qui n'est pas en contact avec la cible) pour être enfin enregistrée.
5. **Transmission, réception et traitement (E)** - L'énergie enregistrée par le capteur est transmise, souvent par des moyens électroniques, à une station de réception où l'information est transformée en images (numériques ou photographiques).
6. **Interprétation et analyse (F)** - Une interprétation visuelle et/ou numérique de l'image traitée est ensuite nécessaire pour extraire l'information que l'on désire obtenir sur la cible.
7. **Application (G)** - La dernière étape du processus consiste à utiliser l'information extraite de l'image pour mieux comprendre la cible.

## Imagerie numérique

### Le « Picture élément » ou pixel

Le pixel est l'élément de base pour la construction d'images. Il porte l'information sur la luminosité d'un point dans une image. C'est la combinaison de ces points avec leurs diverses valeurs de luminosités qui vont définir l'ensemble de toutes les images de l'univers. Ce modèle, facilitera l'exploitation et la manipulation efficace des informations contenues dans les images par l'ordinateur.

La manière d'exprimer la valeur de l'intensité d'un pixel créera différents types d'images, deux valeurs sont considérées, noir (0) et blanc (1). Les images à base de ce genre de pixels sont dites monochromes. Le pixel est codé sur 1 bit.

Plusieurs valeurs sont représentées, et vont modéliser le passage progressif entre les deux couleurs noire et blanche à travers le gris. Généralement nous avons 256 valeurs (1 octet), et les images à base de cette modélisation sont dites de niveaux de gris.

Nous avons rapidement constaté, qu'on peut obtenir toutes les couleurs avec un mélange de trois couleurs de base seulement le rouge, le vert et la couleur bleue (RVB). La modélisation RVB consiste à représenter la couleur d'un pixel avec le mélange des trois couleurs, le codage nécessite 3 octets (un par couleur de base), et ainsi on peut obtenir des images en couleurs.

Une image est une représentation dans le plan d'un objet réel. Les images sont omniprésentes dans notre société (dessins, peintures, photographies, etc.). Depuis quelques années, on assiste à une invasion d'images numériques, aussi appelées images "digitales" par emprunt à l'Anglais ("digit" = nombre).

Ces images sont constituées d'un très grand nombre de très petits points dont la couleur (ou l'intensité) est définie par des chiffres. Ces points sont appelés "pixels", dérivé de l'anglais "Picture éléments".

### Recalage des images (définitions et notations)

Le recalage (ou encore la mise en correspondance) peut se faire entre plusieurs images dites « à recalcer » sur une autre image dite « de référence ou consigne », mais pour simplifier on admettra que le recalage se fait entre une image à recalcer  $I_r$  et une image consigne  $I_c$ .

Et donc, le recalage d'une image sur une image de référence, est l'alignement de l'image à recalcer de sorte que les structures exprimant les mêmes informations soient superposées dans les deux images. Plus précisément, c'est l'optimisation (plus souvent minimisation) d'une fonction exprimant la similarité entre les deux images : image de référence et image recalée.

On définit l'espace des images  $\mathcal{I}$ , l'ensemble des images denses sous forme d'une matrice  $(x, y)$  la dimension en ligne et la dimension en colonne se sont les tailles de l'image, qui contient une densité de niveau de gris compris entre 0 : pour le noir et 255 : pour le blanc et entre les deux bornes c'est les différentes valeurs du niveau de gris alors :

$\mathcal{I} = \{IN \times IN \times IN\}$  pour chaque pixel.

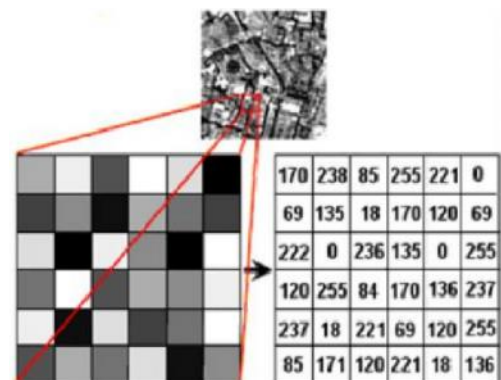


Figure 2. Les images numériques

## Problème de recalage

On peut voir la procédure de recalage comme étant un système représenté par une boîte noire constituée des éléments suivants :

- **En entrée** (essentiellement) une image consigne, et une image à recaler.
- **En sortie** l'image recalée dont la similarité avec l'image consigne est maximale, c'est la fonction objective définie comme suit :

$$\text{MAX Similarité } (I_c, T [I_r]).$$

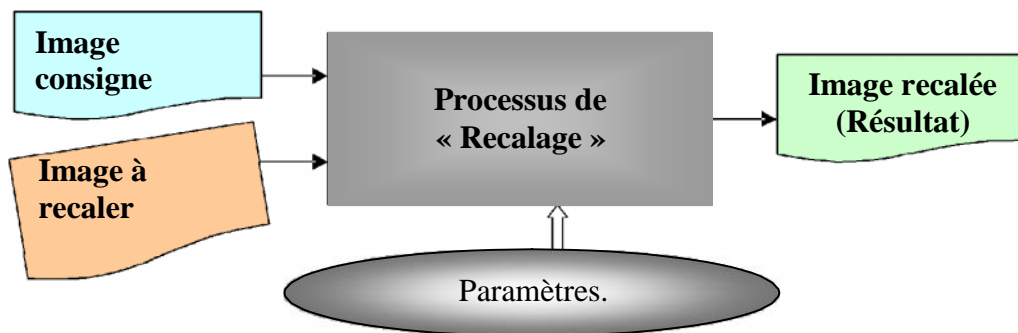


Figure 3. Entrées et sorties de la procédure de recalage.

## Modélisation du Problème de Recalage

Mathématiquement le recalage est le processus de recherche de la transformation  $h$ , (autrement dit c'est l'estimation du champ de déformation  $u(x)=h(x)-x$ ) qui vérifie la relation suivante :

Optimiser  $C(I_c(x), I_r(h(x) = x + u(x)))$ . Pour tout pixel  $x$

Où

ensemble de toutes les images

$C * \rightarrow \mathbb{R}$ .

$(X, Y) \rightarrow C(X, Y) =$  taux de similarité entre  $X$  et  $Y$ .

$h :$   $\rightarrow$

$X \rightarrow h(X) =$  image transformée.

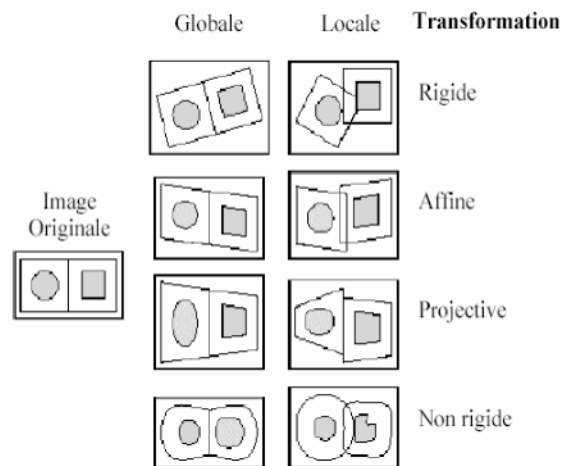
Il existe plusieurs classes de recalage, et comme on ne s'intéresse qu'au cas des images satellitaires les chercheurs proposent de classer ces méthodes selon les critères suivants :

- Les dimensions des images.
- Le type des informations considérées.
- La nature de la transformation considérée et son domaine d'application.
- Interaction machine/homme (approches automatiques, semi automatiques, manuelles).
- La procédure de recherche de la transformation.
- Les modalités (sources d'acquisition) et les natures des images.
- La région (cartes géographiques, images satellitaires et/ou aériennes).
- L'objet de l'image région, zone d'intérêt, ...etc.

## Elasticités de la transformation

Les transformations sont généralement classées dans quatre classes selon les effets qu'elles produisent sur les propriétés géométriques d'une image : *rigide*, *affine*, *projective* et *déformables*.

Une transformation peut être globale ou locale, une transformation globale s'applique sur toute l'image, tandis qu'une transformation locale est un ensemble de transformations appliquées sur des sous parties de l'image en question. Les transformations locales sont presque absentes dans la littérature.



**Figure 4.** Les effets des différentes transformations.

La transformation rigide n'autorise que des translations et des rotations, alors que la transformation affine conserve l'alignement et le parallélisme. Pour ce qui est des transformations projectives, elles permettent la mise en correspondance de lignes non parallèles entre elles et ne conserve l'alignement que pour les droites horizontales ou verticales. Enfin, une transformation déformable permet de transformer des lignes droites en courbes. La figure 4 illustre les diverses transformations en dimension deux. À noter que l'on trouve d'autres classifications des transformations, notamment en deux classes : transformations rigides; transformations non rigides ou élastiques.

### Transformation rigide

Consiste en une rotation et une translation, les distances entre les pixels sont maintenues, les propriétés géométriques (l'alignement et le parallélisme) sont aussi conservées.

La description naturelle d'une transformation rigide est une matrice constante, et le problème revient à trouver un nouvel axe  $y$  à partir de l'ancien axe  $x$  moyennant la translation  $t$  et la rotation  $r$  de l'angle  $\alpha$  :

$$2D \rightarrow y = r.x + t \Leftrightarrow \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

Équation 1 : formule d'une transformation rigide.

Donc le nombre de variables d'une transformation rigide à estimer est de 3 (deux pour la translation et une pour la rotation) dans le cas 2D.

## Transformation affine

Conserve le parallélisme et l'alignement, mais les distances entre pixels ne sont pas maintenues.

En ce qui concerne la description de la transformation affine, on peut garder les mêmes équations linéaires mais sans imposer des restrictions sur la matrice  $r$ , les scalaires de la matrice  $r$  peuvent prendre n'importe quelles valeurs acceptables.

$$2D \rightarrow y = r.x + t \Leftrightarrow \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

Équation 2 : formule d'une transformation affine.

Et le nombre de variables est 6 (en 2D).

## Transformation « projective »

Ne conserve que l'alignement.

## Transformation déformable (ou non rigide)

Où la transformation est libre, elle ne conserve aucune des trois propriétés mentionnées.

La description d'une transformation déformable peut se faire par n'importe quelle fonction sans restrictions, la façon la plus simple est la représentation linéaire où les nouvelles coordonnées se calculent à partir d'une libre combinaison linéaire des anciennes coordonnées.

## Critères de similarité

Plusieurs critères de similarité ont été proposés, le choix d'un critère dépend étroitement de l'approche considérée (dense ou géométrique).

### Critères de similarité pour une approche par primitives géométriques

Dans le cas d'un recalage basé sur une approche par primitives géométriques, on utilise généralement des fonctions exprimant la distance entre deux ensembles de primitives géométriques par exemple la distance euclidienne. Ces primitives nécessitent un prétraitement à faire (prélèvement des primitives communes). Ce genre de similarité peut être très précieux dans le cas de recalage d'images de plusieurs modalités.

### Critères de similarité pour une approche dense

Dans le cas d'un recalage dense on peut envisager plusieurs quantificateurs de similarités reposent sur des formules statistiques mesurant l'interaction entre deux événements tel que: le rapport de corrélation, la covariance, l'entropie ou encore l'information mutuelle [MAE04] pour plus de critères de similarités, ou encore le critère le plus classique, l'erreur quadratique moyenne :

### 1. Erreur quadratique:

$$C(I, J) = \int |I(s) - J(s)|^2 ds = \sum_s |I(s) - J(s)|^2 \quad \text{pour tout pixel } s$$

Équation 3 : formule de l'erreur quadratique.

L'erreur quadratique est toujours positive, et d'autant elle tend vers zéro autant les deux images se ressemblent.

### 2. Entropie:

Opère une seule image  $D$  qui représente la différence entre les deux images  $I, J$ :

$$C(I, J) = H(D) = - \sum_s p(s) \log(p(s)). \quad \text{pour tout pixel } s.$$

$$D = I - J.$$

Équation 4 : formule de l'entropie

La fonction  $p$  décrit la probabilité qu'un pixel ait le niveau de gris  $s$  dans l'image  $D$ .

L'entropie vient de la thermodynamique, elle reflète le taux d désordre d'une masse physique. Elle est toujours positif, et plus elle tend vers zéro le corps est stable.

### 3. Covariance :

$$C(I, J) = COV(I, J) = \sum_s (I(s) - \bar{I} - \bar{J}) \quad \text{pour tout pixel } s.$$

$\bar{I}$  (resp.  $\bar{J}$ ) : les moyennes des niveaux de gris des images  $I$  (resp.  $J$ ).

Équation 5 : formule de la covariance.

Agissant pixel par pixel, la covariance mesure une corrélation de bas niveau entre les deux images. Une covariance nulle reflète une indépendance totale des deux images (pas de similarité), à l'inverse la similarité est maximale dans le cas où la covariance est maximale en valeur absolue.

#### 4. Rapport de corrélation :

$$C(I, J) = R^2(I, J) = \left( \frac{COV(I, J)}{\sqrt{\sum_s (I(s) - \bar{I})^2} \times \sqrt{\sum_s (J(s) - \bar{J})^2}} \right)^2 \quad \text{pour tout pixel } s.$$

Équation 6 : formule du rapport de corrélation.

On peut dire que le rapport de corrélation est l'image standardisée de la covariance, car cette dernière varie dans  $\mathbb{R}$  alors que le premier varie dans  $[0, 1]$ .

Si le rapport de corrélation est nul alors il s'agit d'une totale indépendance des deux images, par contre s'il tend vers 1 alors on peut trouver une combinaison linéaire entre les deux images  $I=a.J+b$ . à ce stade il faut tester encore les hypothèses statistiques  $a=1$  et  $b=0$  pour confirmer la ressemblance parfaite entre les deux images.

#### 5. Information mutuelle :

$$C(I, J) = - \sum_{x,y} p(x, y) \log \frac{p(x, y)}{g(x).h(y)}.$$

pour toute combinaison de niveaux de gris  $(x, y)$ .

Équation 7: formule de l'information mutuelle

$p(x, y)$  est la probabilité (conjointe) qu'un pixel ait le niveau de gris  $x$  dans l'image  $I$  et  $y$  dans  $J$ . la probabilité marginale  $g(x)$  (resp.  $h(y)$ ) est la probabilité qu'un pixel ait le niveau de gris  $x$  (resp.  $y$ ) dans l'image  $I$  (resp.  $J$ ).

L'information mutuelle n'assure pas une combinaison linéaire entre les deux images, mais garantit la co-occurrence de maximum de pixels semblables dans les deux images. Ce qui peut être utile dans le cas de recalage multimodal. L'information mutuelle est maximale en cas de ressemblance d'images, elle est toujours positive.

#### 6. Entropie mutuelle:

$$C(I, J) = - \sum_s p(x, y) \log p(x, y).$$

pour toute combinaison de niveaux de gris  $(x, y)$ .

Équation 8: formule de l'entropie mutuelle

$P(x, y)$  a la même désignation que celle de l'information mutuelle. L'entropie mutuelle est tout le temps positif, minimal en cas de similarité parfaite.

## Chapitre II : Recalage par les métaheuristiques

---

La notion d'optimisation n'est pas récente, les commerçants n'ont pas attendu la programmation linéaire pour essayer de calculer les paramètres optimisant leur gain financier, mais la formalisation mathématique et la création de méthodes de recherche complexes a bien moins d'un siècle. On définit généralement une situation d'optimisation comme le problème consistant à trouver les paramètres à donner à une fonction telle que la valeur retournée par celle-ci soit optimale (minimale ou maximale). On nomme alors fonction objective, ou fonction d'évaluation, la fonction en question. Des contraintes peuvent être imposées pour la recherche de la meilleure solution (espace de recherche, etc.). On peut diviser l'optimisation en deux domaines, selon que le problème soit discret (optimisation combinatoire) ou continu. On pourra aussi trouver des problèmes mixtes, où certaines variables seront discrètes, d'autres continues. La fonction d'évaluation est souvent associée à un problème difficile, dont la solution ne pourrait être trouvée systématiquement avec un algorithme donné en temps raisonnable (problèmes NP-difficiles). Il existe quantité de méthodes pour résoudre ces problèmes, en fonction du besoin (précision, rapidité de calcul, etc.), citons par exemple les méthodes d'approximation réduisant la complexité du problème, la programmation linéaire, ou non-linéaire, les heuristiques spécialisées, ou encore les métaheuristiques, qui sont des méthodes approchées de recherche globale.

Les métaheuristiques ne sont pas des méthodes figées, il n'y a pas de relation d'ordre quant à l'efficacité absolue d'un algorithme ou d'un autre, tout dépend des paramètres utilisés, du problème d'application, d'où l'intérêt de simulations informatisées. En effet, de nombreuses classifications ont été proposées sur les méthodes (ou algorithmes) de résolutions méthodes déterministes/stochastiques; méthodes de recherche locale/globale ; ou encore méthodes exactes/approchées, on s'intéresse aux classes suivantes

### **Les méthodes d'optimisation locale**

Ces méthodes sont sensibles aux optimums locaux, et elles ne permettent pas d'aboutir à une solution proche de l'optimale globale à cause du grand nombre d'optimums locaux, l'algorithme du gradient (la descente amélioratrice « steepest descent ») est l'un de ces techniques, il est condamné à converger vers une solution qui peut être très loin de l'optimale.

### **Les méthodes approchées**

Ces méthodes, ils remédient au problème précédent, ils explorent l'espace de recherche tout en évitant de rester bloqué dans des optimums locaux, ces algorithmes sont de diverses variétés comprenant des techniques stochastiques inspirées des phénomènes naturels, très populaires dans la communauté scientifique (EX Algorithmes Génétiques, Recuit Simulé, etc.). C'est ainsi, que la résolution des problèmes d'optimisation combinatoire, utilise les méthodes approchées, puisque elles évitent les pièges des optimums locaux, et donnent des solutions de très bonne qualité en un temps raisonnable, avec peu d'exigence sur les caractéristiques de la fonction objective comme la continuité et la différentialiste.

En particulier dans ce qui suit, on s'intéresse à ces deux méthodes : l'algorithme génétiques et le recuit simulé qui sont des méthodes de résolution globale approchée.

# Les Algorithmes Génétiques

## Introduction

Un problème d'optimisation est un problème qui se caractérise par une fonction de coût (ou d'énergie) avec ou sans contraintes, défini sur un ensemble fini ou dénombrable de configurations, sur lequel on veut trouver une configuration qui minimise ou maximise la fonction de coût, tout en satisfaisant les contraintes liées au problème. Lorsque le problème est formalisé avec un ensemble de contraintes, et que le domaine de définition de sa fonction de coût est très grand (on ne peut l'énumérer en un temps raisonnable), alors le problème est dit « problème d'optimisation combinatoire », parmi ces derniers on trouve la classe des problèmes les plus difficiles dits NP-Complets.

Résoudre un problème NP-Complet peut être vu comme, un parcours d'arbre dont le nombre de nœuds (un nœud correspond à faire un choix pour une variable de la configuration) croît exponentiellement suivant sa hauteur qui est polynomiale.

Pour des problèmes NP-Complets d'optimisation (tel le problème du voyageur de commerce noté par la suite PVC ou le problème de sac multidimensionnel), on ne connaît pas d'algorithme polynomial permettant de les résoudre de façon optimale, on va donc chercher une solution approchée de cet optimum en utilisant des heuristiques.

En effet, de nombreuses classifications ont été proposées sur les méthodes (ou algorithmes) de résolutions méthodes déterministes/stochastiques; méthodes de recherche locale/globale ; ou encore méthodes exactes/approchées.

L'évolution des espèces a toujours été une des préoccupations des scientifiques, car plusieurs questions subsistaient pour savoir comment certaines des espèces ont disparu et d'autres au contraire ont émergé. Charles Darwin apporte une réponse en 1859 en exposant le concept de sélection naturelle ; et explique que seuls les individus d'une population les mieux adaptés à leurs environnement survivront et se reproduiront plus aisément pour atteindre la prochaine génération. En effet c'est dans une lutte pour la vie que ces individus réussiront à perdurer dans un environnement à l'espace et aux ressources finis, les plus faibles étant condamnés à disparaître.

C'est donc l'aptitude des individus à se reproduire qui est le facteur clé pour la sélection naturelle et leur adaptation à chaque génération. Cette adaptation qui a été par la suite expliquée par Mendel à travers les lois de variation (croisement et mutation).

C'est l'idée d'adaptation des individus les plus puissants et leur survie dans les générations futures qui a inspiré les chercheurs pour résoudre les problèmes d'optimisation dans les différents domaines. Dans les années 1960 et 1970, dès que les calculateurs sont devenus plus accessibles, de nombreuses tentatives de modélisation de l'évolution ont été entreprises. Parmi celles-ci, trois approches ont émergé indépendamment, s'ignorant mutuellement jusqu'au début des années 1990 :

## Description de l'opération naturelle des «algorithmes génétiques» :

Le principe de résolution des problèmes d'optimisation par les algorithmes génétiques est calqué sur le model réel.

En effet, une population d'individus (chromosomes pour faire le lien avec la biologie) sont générées aléatoirement au départ, elles représentent des solutions réalisables du problème et souvent on les code pour des raisons de manipulation (croisement et mutation), les codes utilisés varient en fonction du problème traité (ex : codage binaire, codage en nombres réels, code gray).

L'étape suivante est la sélection des chromosomes qui participent à l'établissement de la génération future, nous le verrons par la suite, une mauvaise sélection peut conduire à une convergence trop rapide vers des optima locaux qui peuvent s'avérer être très éloignés de la solution. Ainsi la sélection est souvent précédée d'une évaluation des chromosomes  $\theta_i$  qui repose sur les résultats que produit chacun d'eux via la fonction objective (fitness  $f(\theta_i)$ ).

Chaque chromosome choisi est ensuite soumis à une probabilité  $P_c$  de croisement avec un autre engendrant ainsi deux nouveaux chromosomes qui ont hérité d'une partie du code génétique de leurs parents, et aussi d'une probabilité  $P_m$  de mutation pour une diversité plus large de la nouvelle population (et couvrir plus de solutions potentielles).

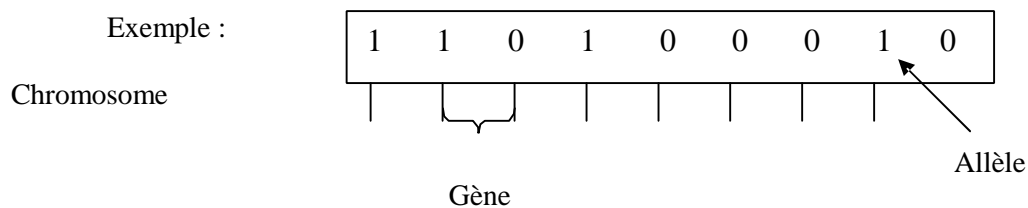
Ces étapes se succèdent dans cet ordre de génération en génération, le critère d'arrêt est soit le nombre de génération soit la convergence de la population vers une même solution (homogénéisation de la population).

Dans ce paragraphe nous expliquons la composition des paramètres utilisés dans l'algorithme génétique et ses différents procédés.

Codage des chromosomes :

Il est évident que l'algorithme dépendra très fortement du type de codage choisi. Le codage se fait généralement sur un nombre de bits fini en binaire, mais cela revient en fait à la nature du problème traité.

Une solution est donc un chromosome formé d'une (ou plusieurs) chaînes de bits où chaque bit représente un gène et son contenu l'allèle.



L'algorithme génère une population initiale  $P_0$  de taille  $n$  qui constituera la base de l'évolution des autres générations, celle-ci se fait de manière aléatoire en tenant compte des contraintes du problème traité et ainsi créer de bonnes solutions de départ et rejeter les mauvaises pour obtenir une population hétérogène et cela a pour but de couvrir l'espace de recherche des solutions.

### Evaluation :

Associe à chaque solution générée son score dans la fonction fitness.

### Sélection :

On distingue 4 types de sélection pour la génération intermédiaire :

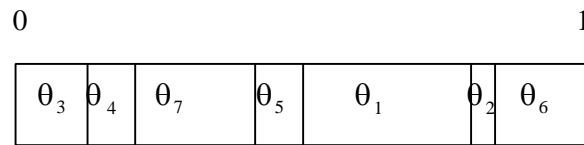
1. Sélection par rang (n/2 élitisme) :

Elle consiste à choisir les n/2 premiers chromosomes ordonnés par ordre décroissant de leurs fitness et compléter la population par la suite à n individus par croisement et mutation. Cette méthode est très contestable car elle conserve les bons individus au détriment des plus mauvais, on remarquera donc une disparition prématurée de quelques solutions (bien que mauvaises) qui aurait peut être contribué à la construction d'une solution optimale. Ce procédé a tendance à converger vers des optima locaux.

2. Roue de la fortune :

Cela revient à faire plusieurs tirages avec remise, l'individu  $\theta_i$  sera retenu avec la probabilité :

$$P(\theta_i) = \frac{f(\theta_i)}{\sum_{k=1}^n f(\theta_k)}$$



Ordonner ensuite ces probabilités sur un segment de droite de longueur 1 puis tirer un nombre  $\alpha \in [0,1]$  et voir à quelle probabilité il correspond, les individus à forte fitness étant ici favorisés, les faibles auront cependant quelques chances d'être sélectionnés.

3. Tournoi :

Ici les chromosomes sont mis par paires au hasard, puis le plus fort est inscrit dans la génération intermédiaire. Dans ce cas aussi, le tirage au sort est fait avec remise.

4. Sélection uniforme :

Chaque chromosome a une probabilité  $P(\theta_i) = \frac{1}{n}$  d'être choisi.

À chaque itération on sélectionne au hasard deux chromosomes dans la génération intermédiaire qui seront sujet à un croisement avec une probabilité  $P_c$  puis une mutation avec une probabilité  $P_m$ .

Croisement :

Il s'opère avec une probabilité  $P_c \in [0.1,0.9]$  en sélectionnant un rang de coupure de façon aléatoire sur les 2 chromosomes choisis, puis interchanger leurs positions. Les points de coupure peuvent bien entendu être augmentés jusqu'à la limite de l'uniformité.

Mutation :

C'est le changement de l'allèle d'un gène sélectionné au hasard avec une probabilité  $P_m \in [0.001,0,01]$  par son complémentaire par exemple.

## **Critère d'arrêt**

Naturellement, on ne peut avoir des nouveaux individus (portant des nouvelles caractéristiques) à partir d'une population homogène dont les individus se ressemblent (i.e. portant les mêmes gènes).

D'une manière analogue, on dit qu'un processus d'optimisation basé sur un algorithme génétique a convergé si la population actuelle est homogène. Une population est homogène si la totalité ou la majorité des individus portent les mêmes gènes (cas discret) ou des gènes dont les valeurs se rapprochent (cas continu).

Le jugement de l'homogénéité d'une population peut s'appuyer directement sur les structures phénotypiques des individus, en testant la ressemblance des gènes, mais cela peut entraîner un coût calculatoire important dans le cas des grands problèmes nécessitant un grand nombre de variables à optimiser. D'où la nécessité de trouver un indicateur, moins coûteux, pour témoigner la convergence de l'algorithme.

Des grandeurs statistiques sur l'évolution des fitness des populations peuvent être chaleureusement adaptés, citons par exemple la moyenne des fitness de tous les individus rencontrés (performance en ligne) ou encore la moyenne des fitness des meilleurs individus par génération (performance hors ligne). Ces grandeurs convergentes vers des valeurs stables suivant la convergence de la population, ainsi on peut juger la convergence de l'algorithme par la stabilité de l'un des deux indicateurs cités précédemment.

## **Réglage des paramètres d'un algorithme génétique**

Tout au long de ce texte, on est arrivé à définir plusieurs paramètres, en plus on a imposé même des contraintes sur certains paramètres (probabilité de croisement qui doit être élevée, et probabilité de mutation faible). Dans ce paragraphe on ne revient pas sur tous les paramètres mais sur quelques uns, en montrant l'influence de chacun sur la performance de l'algorithme.

Commençons par la taille des populations, une taille très faible risque de ne pas créer une diversité, ainsi l'exploration de l'espace sera mauvaise et le processus risque de suivre une convergence prématurée. Tandis qu'une taille très grande risque de ralentir le processus d'optimisation à cause du nombre important d'individus à traiter, ce qui ralentit considérablement la convergence.

En ce qui concerne le taux de mutation, on avait dit que naturellement ce dernier doit être relativement faible, en effet avec une probabilité de mutation élevée les individus auront plus de chance d'être mutés, et comme une mutation a un statut aléatoire, la recherche aura le même statut et favorisera ainsi l'exploration mais pas l'exploitation. De même une probabilité très faible risque de conduire le processus d'optimisation à une convergence prématurée, à cause de la faible exploration de l'espace de recherche.

Un taux de croisement très faible provoque une mauvaise exploration et une mauvaise exploitation.

# Le Recuit Simulé

## Introduction

Le recuit simulé est une version améliorée de la méthode d'amélioration itérative. Il a été proposé en 1983 par Kirkpatrick pour la résolution d'un problème de placement en VLSI (Very Large Scale Integration). La méthode inspirée du principe thermodynamique, dans lesquels les déplacements dans l'espace de recherche sont basés sur la distribution de Boltzmann, cette dernière mesure la probabilité de trouver un système dans une configuration  $X$  d'énergie  $E(X)$ , à une certaine température  $T$  donnée, dans l'espace de configurations  $U$  et elle est définie par

$$P(X) = \frac{1}{Z_T} \exp\left(-\frac{E(X)}{K_B T}\right),$$

$Z_T$  constante de normalisation ;  
Et  $K_B$  constante de Boltzmann.

Dans cette expression, le facteur  $K_B T$  montre que lorsque la température est très élevée, tous les états sont à peu près équiprobables, c'est-à-dire un grand nombre de configurations sont accessibles, par contre quand la température est basse, les états à hautes énergies deviennent peu probables par rapport à ceux de faibles énergies.

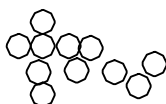
## Description de l'opération physique « recuit » :

En effet, l'état de la matière dépend de la température: à haute température, on a l'état liquide et à basse température, l'état solide.

Si l'on souhaite obtenir un métal avec une structure parfaite de type cristal (c'est l'état fondamental correspondant au minimum d'énergie interne), on va procéder comme suit après avoir porté la matière à l'état liquide, on va abaisser la température jusqu'à solidification. Si la décroissance de température se fait de façon très brusque, on obtient un 'verre', caractéristique de la technique de 'trempe'. Si par contre elle se fait de façon très progressive, laissant le temps aux atomes d'atteindre l'équilibre statistique, on tendra vers des structures de plus en plus régulières pour finir dans l'état fondamental: le cristal, caractérisant le gel du système.

Au cas où cet abaissement de température ne se ferait pas assez lentement, il pourrait apparaître des défauts. Il faut alors les corriger en réchauffant de nouveau légèrement la matière de façon à permettre aux atomes de retrouver la liberté de mouvement, leurs facilitant ainsi un éventuel réarrangement conduisant à une structure plus stable. Ce réchauffement porte le nom de recuit.

État de trempe  
Un verre



état d'équilibre  
une structure parfaite d'atomes



un atome



En s'inspirant de principe de recuit qui est appliqué pour chercher des états stables pour les matières, le recuit simulé cherche à trouver des solutions de très bonne qualité pour un problème d'optimisation.

Donc la matière est vue comme étant une fonction de coût du problème à optimiser, et les états de cette matière peuvent ressembler à des solutions de la fonction en question.

A présent, le processus de recherche peut être vu comme suit

Dans un premier temps, on a une température très haute c'est-à-dire que toutes les configurations de l'espace de recherche sont acceptables, et on va baisser la température lentement pour que les configurations acceptables tendent vers la configuration optimale, la recherche s'effectue par laps, à chaque fois qu'on atteint un état quasi-équilibre on baisse la température, ainsi on va modéliser la recherche pour quelle aboutissent à une solution proche de l'optimale. En effet, contrairement aux autres méthodes (ex. la méthode du gradient, ...), un nouveau trajet de coût supérieur à celui du trajet courant ne sera pas forcément rejeté, son acceptation sera déterminée aléatoirement, en tenant compte de la différence entre les coûts et de la température actuelle.

Le concept de température d'un système physique n'a pas d'équivalent direct dans un processus d'optimisation de type recuit simulé. Le paramètre température  $T$  est simplement un paramètre de control, indiquant le contexte dans lequel se trouve le système (ex: stade de la recherche). En fait, le paramètre  $T$  contrôle les déplacements vers les points voisins les moins bons pour échapper aux optima locaux, sans pour autant trop s'écarter du chemin vers le vrai optimum.

Un critère de voisinage est imposé naturellement, dans un processus de cristallisation du matériel, entre les états. Dans le sens où un laps de refroidissement définit deux états voisins : l'état du matériel avant changement de température, et l'état du matériel après le changement. En gardant l'analogie, les déplacements dans un processus d'optimisation par recuit simulé ne peuvent se faire qu'entre deux solutions voisines.

L'algorithme de Kirkpatrick simule ce processus en combinant dans l'algorithme les mécanismes de refroidissement et de recuit (Figure 5)

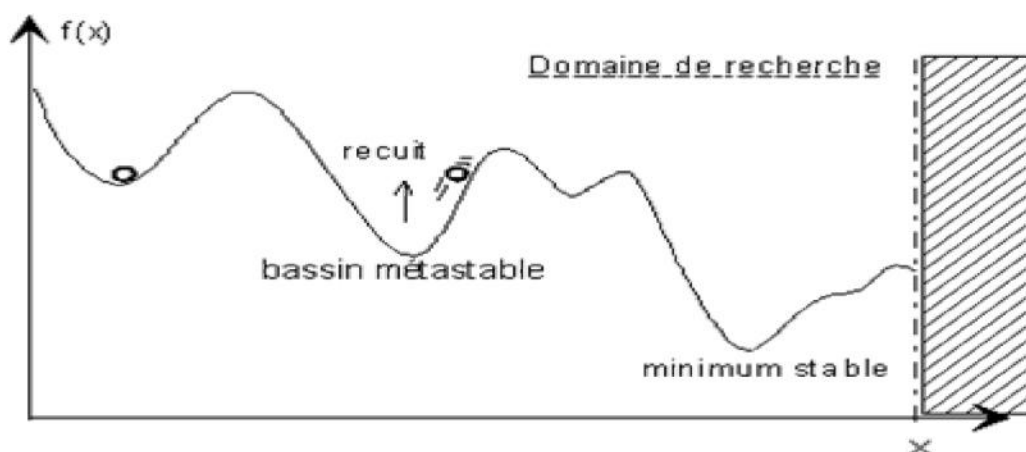


Figure 5. Parcours de l'espace de recherche avec le recuit simulé.

Le principe de "recuit" qui se traduit par une augmentation du niveau d'énergie, permet de sortir des minimums locaux.

En ce qui concerne l'acceptation des nouvelles solutions dans l'algorithme de recuit simulé, la probabilité de Boltzmann n'est pas directement appliquée, mais le critère de Métropolis est utilisé. Le critère de Métropolis permet de décider si une nouvelle configuration générée présente une variation de coût acceptable. Il permet de décider aussi de sortir des minima locaux quand le critère d'arrêt n'est pas encore atteint.

### Critère de Metropolis

À chaque passage d'une configuration X à une configuration Y, on calcule la variation de la fonction de coût  $\Delta f = f(y) - f(x)$ . Le passage est accepté selon la probabilité  $p(X, Y)$  telle que:

$$p(x, y) = e^{\left\{ \begin{array}{c} f(y)-f(x) \\ - \\ T \end{array} \right\}} = e^{\left\{ \begin{array}{c} -\Delta f \\ - \\ T \end{array} \right\}} \quad \text{Equation 9}$$

Lorsque la variation  $\Delta f \leq 0$ , l'exponentielle est supérieure ou égale à 1, la nouvelle configuration doit être acceptée, on lui affecte alors la probabilité maximale de 1. Donc la probabilité  $p(x, y)$  se résume Comme suite

$$p(x, y) = \min(1, e^{\frac{-\Delta f}{T}})$$

Si  $\Delta f > 0$ , on compare  $p(X, Y)$  à un nombre aléatoire  $\mathbf{R}$  dans  $[0,1[$ :

- Si  $\mathbf{R} < p(\mathbf{X}, \mathbf{Y})$  la configuration  $\mathbf{Y}$  est acceptée;
- Sinon elle est rejetée et on essaie une autre configuration.

## Chapitre III : Conception du recalage par les métaheuristiques

Le modèle formel du recalage comme étant le processus de recherche de la transformation qui maximise la similarité entre l'image transformée et l'image consigne.

Donc on propose de voir les choses de la façon suivante :

- **Choix des solutions avec les métaheuristiques et les transformations possibles :** L'image à recaler et l'image de références avec quelques paramètres vont définir le problème de recalage à résoudre.
- **Sélection de la meilleure solution à l'aide des optimiseurs :** Pour ce problème on peut associer un optimiseur de type quelconque. Ce dernier doit permettre à la transformation qui maximisera la similarité entre l'image consigne et l'image à recaler transformée, définie dans le problème en question.
- **Application de la transformation choisie d'une solution sélectionnée:** Une fois la meilleure transformation choisie, on l'applique sur l'image à recaler pour obtenir une nouvelle image recalée dont la similarité avec l'image consigne est maximale.

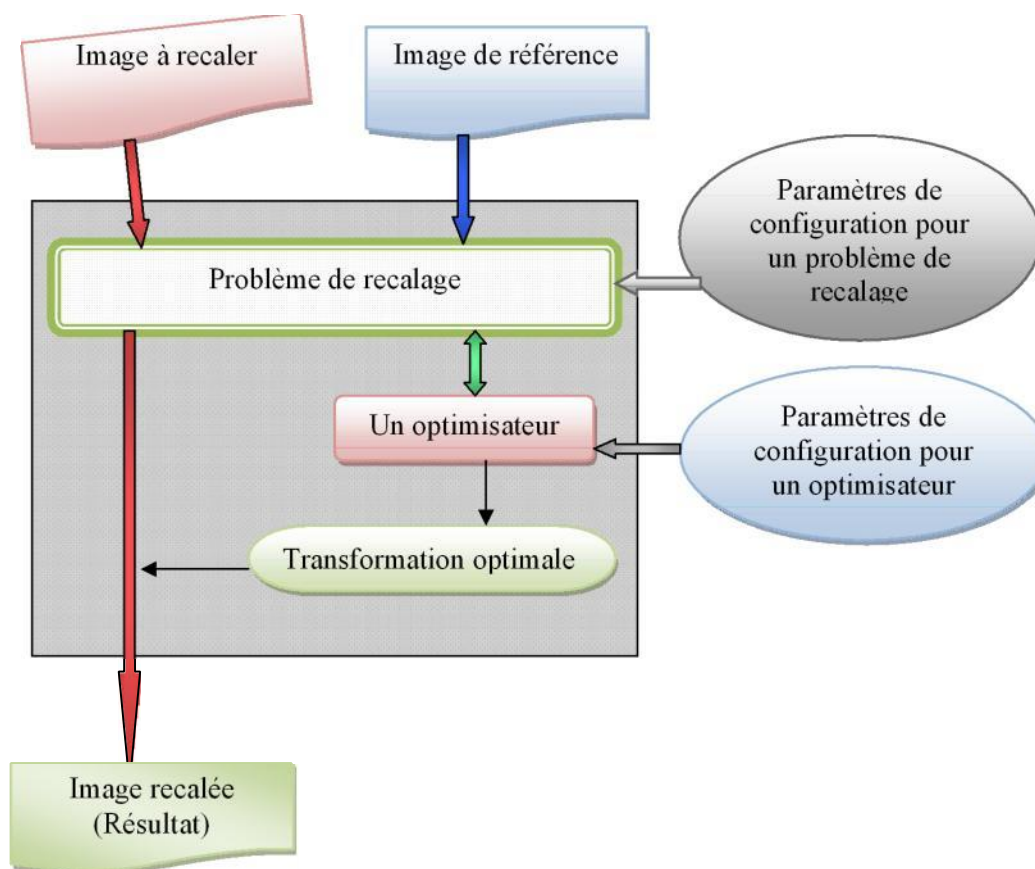


Figure 6. Schéma fonctionnel naturel d'un processus de recalage.

D'après la figure précédente, on constate que la transformation optimale est le seul lien qui relie l'optimisateur et le problème de recalage. L'optimisateur sera considéré comme étant une boîte noire, qui a pour tâche la résolution du problème en entrée et la transformation optimale en sortie.



Figure 7. Entrées et sorties d'une méthode d'optimisation.

Pour la mise en œuvre on a réalisé un logiciel nommé « RIS-MH » : Recalage d'Images Satellitaires, une approche basée sur Méta Heuristiques, on a utilisé le compilateur C++ de Borland C++ Builder 6. On va tirer profit de la modélisation objet qui facilitera la gestion efficace des différentes entités, et de la puissance de calcul d'un compilateur C++.

### Développement du logiciel «RIS-MH»

«RIS-MH» le Recalage des Images Satellitaires par les Métaheuristiques est une plateforme puissante qui a été développé pour

- réaliser la tâche de recalage en utilisant des métaheuristiques
- Valider expérimentalement l'adéquation des métaheuristiques pour résoudre le problème de recalage
- Etudier et analyser les comportements de chaque métaheuristique, (qualité de recalage temps d'exécution totale, coût de chaque partie d'algorithme, ...etc.).

Avec sa conception hautement flexible, grâce à la séparation sémantique des modules qui sont considérés comme étant des boîtes noires avec des entrées et des sorties, «RIS-MH» est toujours prête à recevoir des éventuelles modifications amélioratrices (voir ci-dessous).

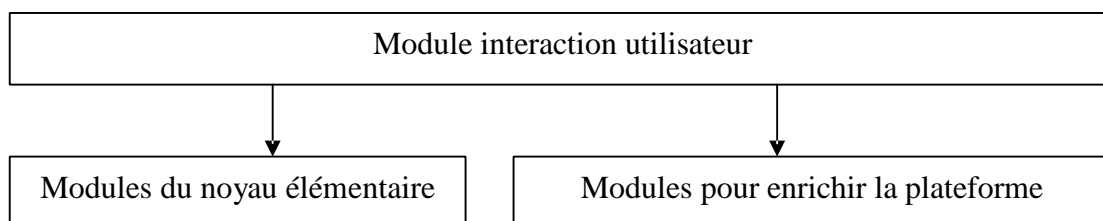


Figure 8. Séparation (sémantique) des modules de la plateforme «RIS-MH».

# Chapitre IV : Accélération du recalage

---

## Introduction

Notre souci est d'améliorer le temps d'exécution du recalage par les méthodes de calcul séquentiels. Ce temps peut être prohibitif quand la taille des images s'accroît. Pour cela, il y a plusieurs méthodes d'accélération des calculs.

Nous allons suivre la stratégie d'Amdahl pour accélérer le processus de recalage dans le sens « make the common case faster » ou en français « rendre le cas le plus commun rapide ». Pour cela, on propose de voir un processus de recalage comme étant la succession de deux parties l'une est l'évaluation des solutions, l'autre englobe la totalité des autres traitements. Nos tests ont donné les résultats suivants

- l'évaluation des solutions est la partie la plus consommatrice de temps de calcul
- Le reste a été consommé par l'ensemble des autres traitements.

Ainsi donc, si on veut accélérer le recalage, il faut se concentrer sur la partie d'évaluation, et pour cela, il faut détailler ce qui se passe durant cette phase. En effet, l'évaluation d'une solution (qui n'est autre qu'une transformation) passe tout d'abord par :

- Le calcul de l'image transformée (cette étape consomme 96% du coût total du recalage),
- Ensuite le calcul de la similarité entre l'image consigne et l'image transformée (qui ne consomme que 03% du coût total).

Ces deux grandeurs nous démontrent que le coût n'est pas emporté par le parcours des pixels (<3%) mais par la complexité des fonctions qui sont nécessaires pour calculer l'image transformée (sinus, cosinus, multiplications,...etc.).

Imaginons cette possibilité, et qu'on peut en avoir une valeur rapprochée du coût d'une transformation dans un problème de recalage, rien qu'en traitant 10% des volumes des deux images, cela entraînera une accélération de 10 relative à l'étape d'évaluation, et une accélération globale de 9.17. Donc on peut diviser le temps total du recalage par 9.17 ce qui nous apporte un gain non négligeable.

## Diminuer la quantité d'information à traiter

Nous avons défini un opérateur d'extraction uniforme déterministe qui avait des résultats similaires aux résultats d'un zoom out des deux images, c'est pour cela qu'on va l'appeler zoom out. Le zoom out de pourcentage P, d'une image dont la hauteur est H et la largeur est W, est une petite image dont le nombre de lignes est  $W' = W \cdot \text{racine}(P/100)$ , et le nombre de colonnes est  $H' = H \cdot \text{racine}(P/100)$ . En effet, les deux images doivent être proportionnelles, cette proposition est modélisée mathématiquement par l'équation,

$$W/H = W'/H'$$

$$\text{Mais } W = \text{nombre\_de\_pixels}/H \text{ et } W' = \text{nombre\_de\_pixels}'/H'$$

$$\text{Et } \text{nombre\_de\_pixels}' = \text{nombre\_de\_pixels} \cdot P/100$$

## Résultats apportés

Nous avons testé la nouvelle méthode de recalage avec le schéma d'accélération décrit auparavant sur les problèmes déjà traités par les méthodes classiques basées sur les algorithmes génétiques et le recuit simulé.

Les tests donnent les résultats suivants

- **Qualité** haute, de même ordre que celle garantie par les approches sans accélération.
- **Temps de réponse** une accélération considérable, l'accélération dépend étroitement du volume des images à traiter et du problème à résoudre. Le tableau suivant résume quelques résultats.

Résolution d'images	Coût calculatoire moyen (AGs)	Coût calculatoire moyen (RS)	Coût calculatoire moyen (AGs+accélération)	Coût calculatoire moyen (RS+accélération)
128*128	50s	5s	07s	01s
256*256	70s	9s	10s	2s
700*800	100s	37s	16s	4s

**Tableau 1.** Comparaison entre les temps de réponse avant et après l'accélération.

## Conclusion

L'analyse des résultats de l'approche de recalage nous a orienté vers l'accélération de la partie d'évaluation des solutions dans le processus d'optimisation (qui est la partie la plus consommatrice). Deux choix étaient possibles pour accélérer cette partie : la parallélisation de calcul, ou la diminution de la quantité des données à traiter.

Dans cette partie nous avons essayé d'accélérer le processus de recalage en diminuant la quantité des données à traiter par un opérateur de zoom out, nous avons remarqué qu'on peut garder l'information utile rien qu'en traitant 5% des deux images, cela entraînera une accélération de 20 relative à l'évaluation des solutions, et une accélération globale de 16.8 (selon Amdahl) qui nous accorde un gain non négligeable. Mais durant les tests et les traitements supplémentaires qu'on a fait, nous avons remarqué qu'en réalité cette approche ne peut optimiser qu'un seul paramètre (l'angle de rotation) tandis que les déplacements avaient toujours de l'écart par rapport à la meilleure solution.

Cela signifié que l'algorithme a convergé vers une mauvaise solution mais très proche de l'optimale, cette solution sera plus tard comme point de départ pour d'autres méthodes d'optimisation plus légère. Cette approche va définir une méthode de recalage automatique rapide et efficace.

# Chapitre V : Parallélisation du recalage

---

## Introduction

Dans le chapitre précédent nous avons vu l'accélération séquentielle des calculs relatifs au processus de recalage à savoir les méthodes d'optimisation que sont le recuit simulé et les algorithmes génétiques. Nous avons vu que nous pouvions réduire considérablement les temps de calcul. Néanmoins, toute autre méthode qui permettra de réduire davantage les coûts de calcul sera étudiée. Dans ce contexte, les architectures parallèles peuvent être d'un grand secours pour la parallélisation d'un certain nombre de calculs.

Nous allons dans ce chapitre faire un tour d'horizon sur les architectures parallèles et voir comment on peut exécuter sur une telle architecture les méthodes d'optimisation qui ont fait l'objet de nos tests. L'accent sera mis sur la partie des calculs qui peuvent et qui doivent être réparties pour obtenir un gain substantiel.

## Parallélisation des algorithmes génétiques

La version séquentielle des AGs a été utilisée avec excellence dans plusieurs applications de divers domaines, mais il existe des problèmes de complexité supérieure et que même les puissantes procédures séquentielles ne peuvent donner des résultats satisfaisants. Donc on cherche toujours à réduire le coût en termes de temps de calcul ou temps de réponse. La parallélisation de traitement est une solution par excellence pour remédier à ce problème. Le processus d'évolution génétiques possède une nature parallèle, l'évaluation et la mutation des individus se fait d'une manière indépendante, les seuls traitements nécessitant une quelque sorte de synchronisation sont la sélection et le croisement.

Les algorithmes génétiques parallèles (AGsP) est donc la solution.

Ce genre d'algorithmes ne représente pas seulement une extension des AGs classiques, mais ils représentent une nouvelle gamme pourquoi pas une nouvelle classe d'algorithmes dont l'exploration est un peu plus particulière, plus différente.

Une bonne parallélisation des algorithmes génétiques doit préserver toutes les caractéristiques des AGs séquentiels, aussi elle ne doit jamais introduire un nombre important des paramètres supplémentaires qui influent sur la qualité de convergence de l'algorithme, finalement elle doit minimiser (éliminer si c'est possible) tout statut de synchronisation, et de communication.

Plusieurs travaux ont été lancés et avaient pour objectif de fournir une classification satisfaisante des AGsP, mais apparemment personne n'a réalisé clairement et pleinement ce travail.

## Parallélisation de l'algorithme du recuit simulé

Différentes possibilités sont offertes pour accélérer la recherche, d'une part on jouant sur les paramètres de l'algorithme (le schéma de la température...), d'une autre part on va faire paralléliser l'algorithme lui même.

## Le Recuit Simulé Evolutionnaire Parallèle (ESA) :

Comme on l'a montré au chapitre précédent, l'« ESA » permet, plus efficacement, d'éviter les pièges des minimas locaux. Donc, lorsque l'algorithme manipule une certaine solution assez

longtemps, il garanti un résultat plus satisfaisant, il a aussi été signalé qu'une population de plus grande taille améliore la performance de la recherche. Cela dit, il parait évident que ces deux dernières propriétés ne soient pas compatibles. C'est-à-dire :

Supposons que nous ayons une population  $P$  et un nombre donné d'évolutions  $N$  où d'évaluations, et soit «  $c_i$  » le nombre sélectionné pour la  $i^{\text{ème}}$  solution. L'idée est d'avoir :  $c_i |P|/N$  pour donner à l'algorithme la possibilité de manipuler suffisamment chaque individu. D'un autre côté, si «  $c_i$  » nécessite «  $t_i$  » par rapport au temps de calcul globale du processus (CPU time), le nombre totale de sélections dans la population «  $c_p$  » et surtout le temps de calcul globale «  $t_p$  » s'exprimeront :

$$c_p = \sum_{i=1}^{|P|} c_i \quad \text{et} \quad t_p = \sum_{i=1}^{|P|} t_i$$

L'opposition entre la taille de la population, qui apporte la diversité, et le temps de calcul peut être résolue par l'utilisation d'une population réduite. Contrairement à l'algorithme génétique, on devra donc travailler sur une population réduite pour que l'avantage de la diversité soit joint au recuit simulé amélioré sans perdre en rapport efficacité/temps de calcul. La réduction de la taille de population détériore généralement le spectre de la diversité. C'est pour palier à ce nouvel obstacle que l'on fait appel à présent à la structure de la programmation parallèle ou distribuée. L'idée est de distribuer le système à travers la distribution d'une population large sur l'ensemble des ressources machines. Pour réussir cela, on doit créer des agents d'identité que l'on exécutera sur les différentes ressources pour chaque sou parti de la population.

Comme on le sait, il existe deux façons d'implémenter un système en exécution parallèle

- La première, en utilisant la parallélisation physique, partitionne l'ensemble des données et les distribue sur des exécutions parallèles sur un ensemble de machines ou de processeurs. La plupart des systèmes évolutionnaires parallèles utilisent cette méthode.
- La seconde, où la parallélisation s'applique à l'algorithme lui-même, est un peu plus compliquée. C'est ce que l'on appelle une parallélisation algorithmique, qui est développé en générale pour des architectures parallèles du type SIMD et MIMD. Etant donnée que la parallélisation algorithmique d'un recuit simulé serait pour le moins très difficile à réaliser, on choisira la parallélisation du système au sens physique.

### **Le Recuit Simulé Evolutionnaire Parallèle sur un système multi agents :**

Pour l'implémentation du « ESA », une architecture de machines parallèles est requise. Les réseaux spécifiques tels que les machines massivement parallèles, n'étant pas accessible en générale, on considérera l'environnement le plus commun et le plus facile d'accès, celui des WAN/LAN (Wide/Local Area Network of computers). La ressource distribuée de machines (DRM) est une infrastructure qui fourni un environnement de résolution de problèmes parallèles basée sur des agents mobiles. C'est l'infrastructure parallèle du logiciel DREAM<sup>2</sup> (Jelasity, Preu et Peachter, 2002), qui à été développée pour une résolution à travers un algorithme évolutionnaire distribué sur un réseau massif de nœuds sur Internet. Le but principal de ce système est la résolution de problème basée sur des systèmes de multi agents, qui exécutent des algorithmes évolutionnaires. Le système est un réseau de ressources qui fonctionne comme un réseau d'échange en nœuds (peer-to-peer) distribué sur une architecture parallèle de machines. Chaque nœud ne possède qu'une connaissance partielle du problème et fonctionne comme le maître de l'ensemble des agents exécutés sur une machine. L'environnement possède donc d'excellentes fonctionnalités pour le

développement de ce type d'application compte tenu de la très bonne communication entre les agents et sa mobilité limitée.

La distribution du processus évolutionnaire sur les ressources d'après DREAM se fait selon de model des îlots (island model). Les îlots sont construits et dotés de plusieurs propriétés, données et algorithmes, puis sont distribués sur le réseau DRM. L'environnement DRM est développé sur la base d'une programmation à traitements multiples en Java (C++).

Les îlots y sont lancés en MIR et sont dotés d'un système de transmissions (Message Passing System, MPS) qui utilise une connexion selon les protocoles TCP/IP. Etant donné que l'« ESA » est d'une nature évolutionnaire, et donc implicitement parallélisable, le DRM permettra facilement de développer l'« ESA » selon un model d'applications sur îlots. Chaque îlot est doté d'un algorithme « ESA » identique et d'une sous partie de population. Il reste à définir le critère de sélection, la règle de remplacement et la stratégie de transfert des résultats atteints par les îlots. Les îlots feront évoluer leur sous population avec leur « ESA » et auront la possibilité de faire migrer une solution vers d'autres îlots selon une certaine règle.

Au sens de la parallélisation, il est préférable d'avoir un temps de calcul global, ( $t_p$ ) réduit pour le même nombre de sélections ( $c_p$ ), ceci pour une seule sélection. Le temps de calcul global pour l'ensemble du processus sera donc réduit linéairement proportionnellement au nombre d'agents  $a$ . Selon les notations utilisées auparavant,  $P$  sera partitionné en  $a$  parties,  $P_1, P_2, \dots, P_a$ . En assignant chaque partition à un îlot, le nombre de sélection étant le même partout, on obtient :

$$c_p = \sum_{j=1}^a c_j, \quad c_j = \sum_{i=1}^{|P_j|} c_i \quad \text{et} \quad t_j = \sum_{i=1}^{|P_j|} t_i$$

Où  $c_j$  et  $t_j$  sont le nombre total de sélection et le temps de calcul total dans l'îlot  $j$ . Alors,  $c_p = a \times c_j$  et  $t_p = \text{Max} \{t_1, t_2, \dots, t_a\}$ .

Donc, pour le même  $c_p$ , on obtient un  $t_p$  plus court, même en considérant le temps de communication entre les îlots. On peut clairement conclure que l'exécution parallèle des agents réduit le temps nécessaire pour atteindre l'optimum global.

# Chapitre VIII : Tests et résultats

---

## Introduction

Tous les éléments ont été collectés pour analyser expérimentalement les résultats qui vont être donnés par le modèle de recalage proposé. En effet, durant cette partie, on va tester les méthodes d'optimisation étudiées, analyser et commenter les résultats apportés par chaque méthode d'optimisation.

En ce qui concerne l'environnement d'exécution, tous les tests ont été effectués sur un ordinateur personnel dont les caractéristiques sont :

- Matériel processeur Intel Pentium 4 (1GHz), 256Mo de RAM.
- Système d'exploitation Microsoft Windows XP service pack2.

Etant donné que le coût de calcul fut parmi les paramètres critiques qui valident la performance d'une solution informatique implémentée pour résoudre un problème d'optimisation, et pour analyser correctement la relation entre les différentes entités et le coût relatif à chacune, nous n'avons considéré que le temps consommé par la méthode d'optimisation et ses entités. Les résultats de ce chapitre ne comptabilisent pas le coût des traitements des entrées/sorties.

## Définitions

- Les paramètres génétiques (ou une configuration génétique) sont les paramètres d'un algorithme génétique le critère de fin de recherche, l'opérateur de sélection, de croisement, le taux de mutation, et évaluation ainsi le critère de similarité...etc.
- Les paramètres recuit (ou une configuration recuit) c'est l'ensemble des paramètres d'une méthode d'optimisation recuit simulé.
- Une configuration (génétique ou recuit) **perdante** c'est une configuration qu'on a peu de chances de converger vers des bonnes solutions, contrairement avec la configuration **gagnante**, on a la chance de converger vers de très bonnes solutions
- Une configuration **intermédiaire**, il y a de grandes chances de converger vers des résultats moyens (pas bons et pas mauvais), elle est comme une **position stratégique**, à partir de laquelle on peut constater les améliorations et les diminutions de performances du recalage suite à une modification des paramètres.

## Méthodologie des tests

Pour être cohérents et efficaces, nous avons suivi les étapes suivantes pour valider un modèle de recalage à base d'une métaheuristique

- Choix des paramètres optimaux pour chaque méthode d'optimisation afin d'assurer une configuration gagnante.
- Adopter ces paramètres pour résoudre les problèmes de recalage réels.

## Choix des paramètres optimaux

Les deux méthodes étudiées et proposées sont des méthodes paramétriques, dont le choix des paramètres ne se fait que par « tâtonnement » (i.e. faire beaucoup de tests pour estimer une valeur adéquate pour chaque paramètre), notre stratégie consiste à faire un maximum de tests sur des problèmes dont on connaît le meilleur résultat.

Cette étape va se dérouler en deux parties : la recherche des paramètres optimaux, ensuite la validation du choix.

## La recherche des paramètres optimaux

On va essayer d'étudier la variation de performances (coût de calcul et qualité) par rapport à la variation des paramètres de la méthode d'optimisation considérée. Il faut dire qu'à ce stade, les méthodes d'optimisation adoptées suivent des schémas stochastiques (i.e. il y aura forte chance de converger vers des résultats différents après deux exécutions consécutives), donc il est judicieux d'estimer une probabilité de convergence vers une certaine classe de résultats, cette approche nous permettra un meilleur contrôle des tests et une meilleure interprétation des résultats.

Cette étape consiste en un ensemble de tests qui ont pour but la recherche des paramètres optimaux de chaque méthode, ces tests vont se dérouler (pour résoudre le même problème de recalage avec les mêmes images et la même transformation à chercher) la façon suivante :

- Prendre une configuration intermédiaire.
- Varier un seul paramètre à la fois, pour chaque valeur lancer 20 fois le processus de recalage.
- Etudier les résultats
  - **Evaluer la qualité** de recalage.
  - **Estimer la probabilité** de convergence vers une certaine classe de résultats.
  - **analyser le coût** en temps de calcul machine.

## Conclusion générale & perspectives

---

Le recalage des images reste d'une nécessité importante surtout dans le domaine de la télédétection, vu les applications de diagnostics et thématiques très nombreuses de haute importance pour les géomaticiens, les géophysiciens et les géographes qui peuvent être facilitées considérablement moyennant le recalage. Ce dernier est aussi une tâche difficile relativement à la quantité importante des informations à traiter d'une part, et la nature de paysage de la fonction à optimiser (qui est fortement non linéaire) d'autre part. Et entre ces deux points les recherches se sont orientées essentiellement vers la modélisation qui répond au mieux aux attentes des cadres de la télédétection. Et malgré les efforts, apparemment il n'existe pas une approche permettant de répondre aux attentes, qui sont toutes simples une bonne déformation en un temps particulièrement limité.

Par exemple les modèles élastiques non linéaires et fluides produisent des résultats de très bonnes qualités, mais le coût calculatoire d'optimisation de la déformation est trop élevé. Donc ces deux approches restent prohibées jusqu'à nouvel ordre.

Le recalage d'image étant la maximisation de la ressemblance d'image, il devient ainsi l'un des problèmes d'optimisation qui nécessite une méthode d'optimisation adéquate.

Les métaheuristiques étant les meilleures méthodes d'optimisation, grâce à leurs stratégies intelligentes de recherche, peuvent être, théoriquement, adéquats au problème de recalage. C'est pour cela que nous avons étudié l'adéquation théorique de deux métaheuristiques (algorithmes génétiques et recuit simulé) avec tous les schémas de codage possibles (naturel, binaire ou quantique).

Pour valider le modèle (recalage par métaheuristiques), nous avons mis au point «RIS-MH» une plateforme pour le Recalage d'Images Satellitaires basée sur Métaheuristiques. «RIS-MH» avec sa grande flexibilité, permet l'intégration de n'importe quelle méthode d'optimisation pour le recalage, et grâce à la séparation des contextes « optimisation/recalage » «RIS-MH» augmentera également la réutilisation de ses bibliothèques.

Le module d'interface utilisateur de «RIS-MH», permet le pilotage et le contrôle de la totalité des paramètres des différentes entités : problèmes de recalage, configurations génétiques, configurations recuit, type de codage, critères de similarités, etc. Cela nous donne l'aisance de faire des tests significatifs : un choix optimal des paramètres de chaque méthode, des comparaisons selon la performance entre les différentes méthodes, et surtout les résultats précieux qui ont été tirés.

La robustesse de la méthode algorithmes génétiques au problème de recalage est meilleure que celle de la méthode recuit simulé, mais la performance de la première n'était pas à la hauteur de la seconde. Le recuit simulé nous a accordé les mêmes qualités de recalage dans un temps très réduit relativement en temps consommé par les processus basés sur algorithmes génétiques.

Nous avons constaté, que l'évaluation était la partie la plus consommatrice dans les deux schémas d'optimisation, ce qui nous a guidé vers un schéma d'accélération basé sur la minimisation de la quantité d'informations à traiter. Le schéma accéléré, basé sur le recalage d'images zoomées, n'a pas accordé de bonnes solutions au problème de recalage, mais il a permis de trouver une solution relativement proche de l'optimale. Cette solution a été utilisée comme un bon point de départ pour une autre méthode. Et trouver un bon point de départ conduit à la rapidité de convergence, et la diminution considérable du temps de calcul. A la fin, nous avons proposé une méthode de recalage automatique basé sur un schéma accéléré des méthodes d'optimisation dont les paramètres optimaux ont été fixés sur les valeurs identifiées durant l'étape du choix des paramètres optimaux.

## Bibliographie

---

[AND99] P. Andrey « **segmentation d'images par algorithmes génétiques** » thèse doctorat, sciences appliquées, paris 7, 1999.

[BEL99] Belaji Natrajan and bruc E. Rosen « **imagine enhancente using very fats simulated annealing** » division of cumputer science, The university of texas at san antonio,IEEE. 1999.

[BEL03] Silva, L.; Bellon, O.R.P.; Gotardo, P.F.U.; Boyer, K.L. “**Range image registration using enhanced genetic algorithms**”[Image Processing, 2003. ICIP 2003. Proceedings. 2003. International Conference on](#) Volume 2, 14-17 Sept. 2003 Page(s):II - 711-14 vol.3 Digital Object Identifier 10.1109/ICIP.2003.1246779

[BEN03] Benoît Rulleau « **simulation arithmétique et optimisation de la structure des zones de qualification** » thèse, laboratoire d'optimisation globale, centre d'études de la navigation Aérienne, Toulouse. 18 septembre 2001.

[BON95] Christophe Bontemps « **Principes Mathématiques et Utilisations des Algorithmes Génétiques**» 18 Novembre 1995, URL <http://www.toulouse.inra.fr/centre/esr/CV/bontemps/WP/AlgoGene.htm>

[CHA88] R.D. Chamberlain, M.N. Edelman, M.A. Franklin and E.E. Witte. « **Simulated annealing on a multiprocessor**” [Computer Design: VLSI in Computers and Processors, 1988. ICCD '88. Proceedings of the 1988 IEEE International Conference on](#) 3-5 Oct. 1988 Page(s):540 - 544

[CHR01] G.E Christensen, H.G Johnson « **Consistent image registration**”[Medical Imaging, IEEE Transactions on](#) Volume 20, Issue 7, July 2001 Page(s):568 - 582

[CHM03] L. Chielewski and D. « **Image registration**”[Medical Imaging, IEEE Transactions on](#) Volume 22, Issue 11, Nov. 2003 Page(s):1341 - 1343

[GLA01] Glassner, A.; « **Quantum computing. 3** » [Computer Graphics and Applications, IEEE](#) , Volume 21, Issue 6, Nov.-Dec. 2001 Page(s):72 – 82

[GOL00] Marin Golub and Domagoj Jakoviè “**A new model of global parallel genetic algorithm**”[Information Technology Interfaces, 2000. ITI 2000. Proceedings of the 22nd International Conference on](#) 13-16 June 2000 Page(s):363 - 368

[DRA04] A. DRAA, H. TALBI, M. BATOUCHE « **Une approche génétique quantique pour la fusion d'images multi-sources** » Equipe vision et infographie, Laboratoire LIRE Université Mentouri Constantine ALGERIE, 7/10/04

[HAR98] Harik, G.R.; Lobo, F.G.; Goldberg, D.E.; « **The compact genetic algorithm** »[Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, the 1998 IEEE International Conference on](#) 4-9 May 1998 Page(s):523 - 528

- [HOL75] J.H. Holland “**adaptation in natural and artificial systems**” university of Michigan press, 1975
- [HUE03] R.H. Huesman, G.J. Klien, J.A.Kimdon, C.Kuo, S.Majumdar « **Deformable registration of multimodal data including rigid structures**” [Nuclear Science, IEEE Transactions on](#) Volume 50, Issue 3, Part 2, June 2003 Page(s):389 - 392
- [ISS 04] Hazem ISSA "Mise en correspondance des stéréoscopique par algorithmes génétiques: nouveaux codages" thèse Doctorat Lille 1 2004.
- [JON75] K. De Jong “**an analysis of the behavior of a class of genetic adaptative systems**” PhD thesis, University of Michigan, 1975
- [JOH03] D. Johan et al. « **Métaheuristiques pour l’optimisation difficile** » Edition Eyrolles, 2003
- [KIR83] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi « **optimization by simulated annealing** » article Science volume 200, Numéro 4598. 13 mai 1983, URL <http://citeseer.ist.psu.edu/kirkpatrick83optimization.html>
- [KUK02] Kuk-Hyun Han; Jong-Hwan Kim; « **Quantum-inspired evolutionary algorithm for a class of combinatorial optimization**” [Evolutionary Computation, IEEE Transactions on](#) Volume 6, Issue 6, Dec. 2002 Page(s):580 - 593
- [LI 04] Li, W.; Leung, H.; “**A maximum likelihood approach for image registration using control point and intensity**” [Image Processing, IEEE Transactions on](#) Volume 13, Issue 8, Aug. 2004 Page(s):1115 - 1127
- [LU 91] Lu, N.-A.; Morrell, D.R.; « **VQ codebook design using improved simulated annealing algorithms**” [Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on](#) 14-17 April 1991 Page(s):673 - 676 vol. 1
- [MAE03] Yoichiro Maeda and Toru Tsubouchi “**parallel genetic algorithm used fuzzy adaptative search method**” SICE annual conference in Fukui, August 4-6, 2003.
- [MAE04] Yoichiro Maeda and Masahide Ishita “**fuzzy adaptative search method for parallel genetic algorithm with combined sub-populations**” proceeding of the 2004 IEEE conferences 25-29 July, 2004.
- [MAT 02] Matthieu D. "les algorithmes génétiques" 2002.
- [MEN97] Mendonca, P.R.S.; Caloba, L.P.; « **New simulated annealing algorithms** » [Circuits and Systems, 1997. ISCAS '97. Proceedings of 1997 IEEE International Symposium on](#) Volume 3, 9-12 June 1997 Page(s):1668 - 1671 vol.3
- [MOI98] Le Moigne, J.; Wei Xia; Chalermwat, P.; El-Ghazawi, T.; Mareboyana, M.; Netanyahu, N.; Tilton, J.C.; Campbell, W.J.; Crompton, R.P.; First “**evaluation of automatic image registration methods**” [Geoscience and Remote Sensing Symposium Proceedings, 1998. IGARSS '98. 1998 IEEE International](#), Volume 1, 6-10 July 1998 Page(s):315 - 317 vol.1

[LAH07] Lahlou BENGHEZAL et Rachid OUAFI « Satellite Images Registration by Metaheuristics. » communication nationale dans le cadre de la première Journée Nationale sur les Applications des Métaheuristiques (JNAM'07), le 29 mai 2007 à la Faculté d'Electronique et d'Informatique de l'U.S.T.H.B Alger.

[PLU03] J.P.W. Pluim and J.M. Fitzpatrick "image registration" [Medical Imaging, IEEE Transactions on](#) Volume 22, Issue 11, Nov. 2003 Page(s):1341 - 1343

[REI98] E. Reiffel, W. Polak « An introduction to quantum computing for non-physicists » august 16, 1998.

[ROU98] J.-M. Rouet, J.-J. Jacq and C. Roux « 3D elastic multimodality image registration through a genetic algorithm » [Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE](#) Volume 2, 29 Oct.-1 Nov. 1998 Page(s):663 - 666 vol.2

[SAL01] M. SALOMON « Étude de la parallélisation de méthodes heuristiques d'optimisation combinatoire application au recalage d'images médicales » thèse, université Louis Pasteur 11 décembre 2001.

[SOO96] Soo-Young Lee; Kyung Geun Lee; "Synchronous and asynchronous parallel simulated annealing with multiple Markov chains » [Parallel and Distributed Systems, IEEE Transactions on](#) Volume 7, Issue 10, Oct. 1996 Page(s):993 - 1008

[SOU 04] Souquet A. et Radet Francois-Gérard "Algorithmes génétiques " 2004.

[SUN04] Youfa Sun and Feiqi Deng « Chaotic parallel genetic algorithm with feedback mechanism and its application in complex constrained problem » [Cybernetics and Intelligent Systems, 2004 IEEE Conference on](#) Volume 1, 1-3 Dec. 2004 Page(s):596 - 601 vol.1

[TAL04] Talbi, H.; Draa, A.; Batouche, M.; "A new quantum-inspired genetic algorithm for solving the travelling salesman problem" [Industrial Technology, 2004. IEEE ICIT '04. 2004 IEEE International Conference on](#) Volume 3, 8-10 Dec. 2004 Page(s):1192-1197 Vol. 3

[VAL01] Thomas Vallée et Murat Yildizoglu « présentation des algorithmes génétiques et de leurs applications en économie » V. 1.2. , 07 septembre 2001. URL [beagle.u-bordeaux4.fr/yildi/files/agpresf.pdf](http://beagle.u-bordeaux4.fr/yildi/files/agpresf.pdf)

[WAH00] Benjamin W. Wah and Yi Xin Chen « optimal anytime constrained simulated annealing for constrained global optimization\* » department of Electrical and computer engineering and coordinated science laboratory, University of Illinois, urbana-champaign, 1308 West main street Urbana, USA, URL <http://www.manip.crhc.uiuc.edu>

[YVE01] Yves Coueque - Julien Ohler - Sabrina Tollari « Algorithmes génétiques pour résoudre le problème du commis voyageur » avril 2001 URL <http://sis.univ-tln.fr/~tollari/TER/AlgoGen1>

[ZIV03] B. Zivota and J. Flisser "image registration methods: a survey" image and vision computing, vol. 21, pages: 977-1000, June 2003.