

2.1 Introduction

L'observation des phénomènes biologiques est une très riche source d'inspiration pour les informaticiens. Le concept d'algorithme génétique notamment, généralisé ces dix dernières années sous le terme d'algorithme évolutionnaire en est un excellent exemple. Les principes de base de ces algorithmes sont une transposition informatique simplifiée de la très célèbre théorie de Darwin. En d'autres termes, on imite au sein d'un programme la capacité d'une population d'organismes vivants à s'adapter à son environnement à l'aide des mécanismes de sélection et d'héritage génétique. Depuis une quarantaine d'années de nombreuses méthodes de problèmes d'optimisation stochastiques, ont été développées à partir de ces principes, pour constituer ce que l'on nomme maintenant le « Darwinisme artificiel ». Le terme : algorithmes évolutionnaires, couvre en fait un ensemble de techniques, nommées : Algorithmes génétiques, programmation génétique Stratégies d'évolution, programmation évolutionnaire, suivant la façon dont les principes Darwiniens sont traduits dans le modèle artificiel. Ces derniers seront l'objet de ce présent chapitre, [7].

2.2 Historique

Le calcul évolutionnaire a été introduit dans les années 1960 par I. Rechenberg dans son travail « Evolution strategies ». Son idée a été développée par les autres chercheurs.

Les algorithmes génétiques ont été inventés par John Holland et ont été développés par lui et ses étudiants et ses collègues. Il introduit le premier modèle formel des algorithmes génétiques dans son livre « Adaptation in natural and artificial Systems » publié en 1975.

En 1992, John Koza a utilisé l'algorithme génétique pour évoluer des programmes pour exécuter certaines tâches. Il a appelé sa méthode « genetic programming » (GP), [7].

2.2.1 Présentation

L'algorithme génétique, comme l'indique son nom, est un algorithme itératif qui simule les mécanismes de reproduction des organismes vivants pour trouver une solution à un problème donné. La recherche des solutions s'effectue par des opérations comme la reproduction, le croisement et la mutation. Parmi ses avantages, la robustesse et la possibilité d'application sans connaître a priori les propriétés du domaine de recherche.

Comme pour beaucoup d'algorithmes, les paramètres de l'algorithme génétique doivent être soigneusement choisis pour obtenir de bons résultats. Souvent, ces paramètres dépendent du système, [8].

2.2.2 Caractéristiques

Les Caractéristiques des algorithmes génétiques peuvent se résumer par les aspects fondamentaux suivants :

- ❖ Les AG travaillent sur un codage de l'ensemble des paramètres eux-mêmes.
- ❖ Les AG cherchent l'optimum à partir d'une population de points et non à partir d'un seul point.
- ❖ Les AG utilisent l'information (ou le coût) de la fonction objectif et non pas les informations provenant des fonctions dérivées de quelque ordre que ce soit.
- ❖ Les AG font appel à des règles de transition probabilistes et non pas à des règles déterministes, [8].

2.2.3 Vocabulaire utilisé par les algorithmes génétiques

Pour faire le parallèle entre la nature et les AG, on spécifie Les termes suivants :

Nature	Algorithme génétique
Chromosome	Chaîne de gènes
Gène	Trait, caractéristique, ou détecteur
Allèle	Valeur de la caractéristique
Locus	Position de la chaîne
Génotype	Structure (ensemble des gènes représentés Par un chromosome)
phénotype	Ensemble des paramètres
Epistasie	Non linéarité

Tabl. 2.1 Vocabulaire utilisé par les algorithmes génétiques, [8].

2.3 Principe de fonctionnement des AGs

Les AG commencent avec une population de créatures artificielles (chaînes de caractères) analogue aux chromosomes en nature qui évoluent par application de plusieurs opérateurs (modélisant le test de la survie naturelle) vers la solution optimale ou du moins vers une solution proche de celle-ci.

Une solution (un chromosome) résout un problème, si elle satisfait une fonction de performance donnée. L'ensemble des solutions est appelé espace des solutions.

L'objectif de la procédure de recherche est de localiser une solution satisfaisante.

Une description abstraite de l'AG de base peut se faire selon les étapes suivantes :

- ❖ Initialiser aléatoirement une population de chromosome (individus).
- ❖ Evaluer chaque chromosome qui est associé à une fonction coût ou fonction d'aptitude déterminant son rang dans la population. Cette fonction est l'arbitre final décidant la vie ou la mort de chaque individu.
- ❖ Créer de nouveaux chromosomes en appliquant les opérateurs de la sélection et de la reproduction.
- ❖ Evaluer les nouveaux chromosomes (les descendants) et les insérer dans la population pour construire une nouvelle génération.

Ce processus se répète jusqu'à la satisfaction du critère d'AG qui est généralement spécifié par un nombre de générations, [7].

2.3.1 Structure d'un AG

L'organigramme fonctionnel suivant, illustre la structure Générale d'un AG standard :

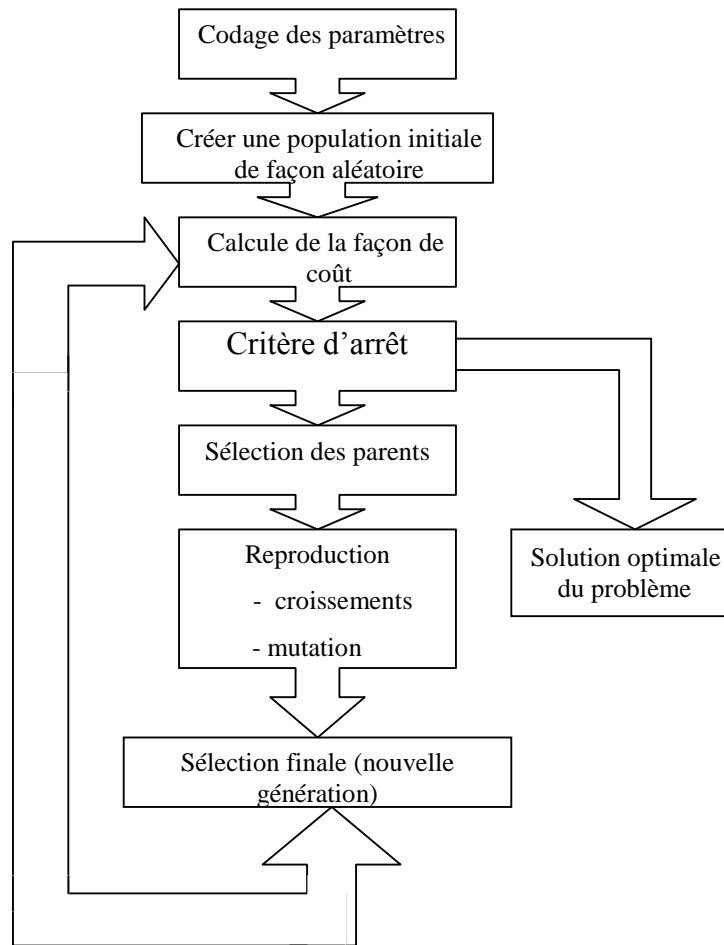


Figure. 2.1 Organigramme d'un AG Standard

2.3.2 Comparaison avec d'autres méthodes

En se basant sur le principe de fonctionnement des AG, on identifie quelques différences avec d'autres méthodes classiques d'optimisation :

- ❖ Les AG cherchent une représentation codée dans l'espace des solutions et non pas directement dans le domaine original.
- ❖ Les AG utilisent un espace de recherche plus vaste, limité par la taille de la population.
- ❖ Les AG utilisent des règles de transition probabiliste et non déterministes (pseudo aléatoires).
- ❖ Les AG n'utilisent que les valeurs de la fonction à optimiser, pas sa dérivée ou une autre connaissance auxiliaire, [7].

2.4 Mécanisme de fonctionnement des AG

2.4.1 Codage des variables

La première étape qui est importante est de définir et de coder convenablement le problème, à chaque variable est représenté par un individu doté d'un génotype constitué d'un ou plusieurs chromosomes. Nous appelons populations un ensemble de N individus que nous allons faire évoluer. Il y a plusieurs formes de codage (codage réel, codage de gray ...), on note : les différents type de Codage

❖ **codage binaire :**

Un gène est un entier long (32 bits), un chromosome est un tableau de gène....

10010011	11101011	0011010
$X_1 = 3.256$	$X_2 = 0.568$	$X_3 = 10.26$

Figure. 2.2 Exemple du codage

L'avantage du codage binaire est que l'on peut facilement coder toutes sortes de variable : des réels, des entiers, des valeurs booléennes, des chaînes de caractères....

Espace de recherche :

Dans certains algorithmes d'optimisation, tels que les stratégies de évolution, l'espace de recherche est infini : seule la population initiale est confinée dans un espace fini, mais dans le

cas des AG, il est généralement nécessaire de définir un espace de recherche fini. En effet, cette limitation ne serait que pour des raisons informatiques, les intervalles de définition des variables sont en général naturellement limités comme suit :

$$X_i \min \leq X_i \leq X_i \max \quad \forall i \in [1, n]$$

Afin de coder nos variables réelles en binaire (un coda sur 32 bits), nous Discrétisons l'espace de recherche $g_{\max} = 2^{32} - 1 = 4294967295$ valeurs discrètes.

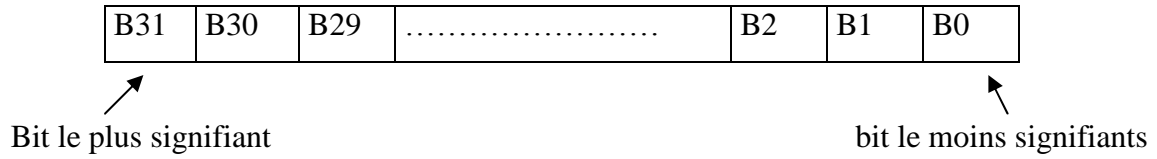


Figure. 2.3 Codage binaire

A chaque variable réelle X_i , on associe donc entier long G_i :

$$0 \leq g_i \leq g_{\max} ; \quad \forall i \in [1, n]$$

Les formules de codage et décodage sont alors suivantes :

$$g_i = \frac{x_i - x_{i \min}}{x_{i \max} - x_{i \min}} g_{\max} \quad (2-1)$$

$$x_i = x_{i \min} + \frac{(x_{i \max} - x_{i \min})}{g_{\max}} g_i \quad (2-2)$$

Codage par permutation de valeurs entières :

Ce codage est bien adapté à l'ordonnancement.

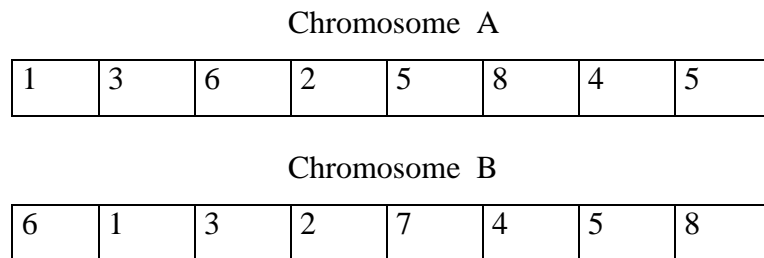


Figure. 2.4 Codage par permutation de valeurs entières

❖ Codage par valeurs réelles :

Ce codage est généralement utilisé pour des valeurs qu'on ne peut pas mettre sous la forme d'un des deux codages précédents. Ces valeurs sont fortement liées au problème à résoudre.

Chromosome A							
A	A	C	G	T	T	C	G

Chromosome B							
2.34	5.67	0.50	0.34	1.87	5.78	3.67	9.12

Chromosome C							
Bleu	Rouge	Vert	Bleu	Bleu	Rouge	Rouge	Noir

Figure. 2.5 Codage par valeurs réelles

On aboutit à une structure présentation les cinq niveaux d'organisation des paramètres d'un AG :

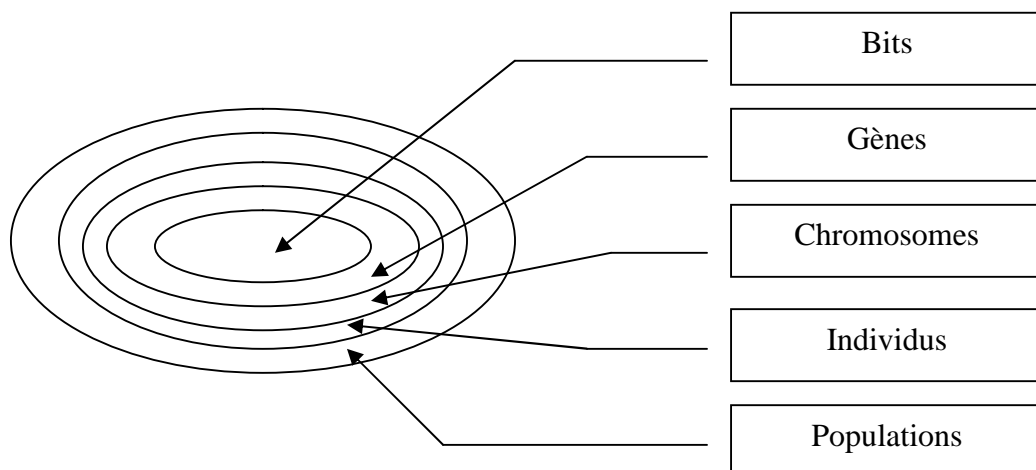


Figure. 2.6 Les cinq niveaux d'organisation des paramètres d'un AG, [7].

2.4.2 Fonction d'adaptation

La fonction d'adaptation, ou fitness, est une fonction à optimiser, qui associe une valeur pour chaque individu. Cette valeur a pour but d'évaluer si un individu est mieux adapté qu'un

autre à son environnement, ce qui signifie qu'elle quantifie la réponse fournie au problème pour une solution potentielle donnée, ainsi les individus peuvent être comparés entre eux.

Cette fonction, propre au problème, est souvent simple à formuler lorsqu'il y a peu de paramètre. Au contraire, l'ors qu'il y a beaucoup de paramètre ou l'ors qu'ils sont corrélés, elle est plus difficile à définir, dans ce cas, la fonction devient une somme pondérée fonction, un ajustement des coefficients est alors nécessaire, [7].

❖ Importance du fitness

L'évaluation du fitness permet une classification des individus dans la population. Le critère utilisé dépend du problème spécifique à résoudre. Le mécanisme qui est utilisé pour améliorer le fitness des individus constitue la plus importante partie de la procédure. Dans la phase de reproduction, des paires d'individus qui sont destinées à croiser doivent être sélectionnées de telle sorte que la probabilité de sélection est plus importante pour les individus ayant un fort fitness, [7].

2.4.3 Evaluation

L'évaluation de chaque dispositif est réalisée par le modèle utilisé. Si le dispositif possède plusieurs états de fonctionnement, le modèle peut être lancé plusieurs fois. Les résultats obtenus sont alors utilisés pour calculer les fonctions objectif et la fonction d'adaptation.

Pour appliquer les AG, on doit avoir une fonction de coût (d'évaluation) G à minimiser indirectement. Sa valeur est élaborée en une valeur d'aptitude F qui doit être maximisée. Chaque chromosome peut représenter une différente chaîne de caractères et une idée différente sur la résolution du problème, satisfaisant ainsi la fonction d'évaluation. La décision du codage des chromosomes représenter les solutions possibles d'un problème particulier mène à l'exigence d'un choix convenable de la fonction d'évaluation. Cette dernière doit être capable d'interpréter les données contenues dans les chromosomes et de décider si la solution résultante est optimale ou non. La fonction d'évaluation prend un chromosome et retourne une valeur qui lui est associée la fonction coût G et calculée et transformée pour qu'elle convienne dans un AG.

En raison de son analogie avec le processus de l'évolution naturelle, l'AG est naturellement en termes de maximisation, [7].

2.4.4 Opérateur de reproduction

Les AG sont basés sur un phénomène naturel qui est l'évolution, plus précisément, ils supposent, qu'à priori, deux individus adaptés leur milieu donne, par recombinaison de leurs gènes, des individus mieux adaptés. Pour ce faire, trois opérateurs sont à disposition : la sélection, le croisement et la mutation.

❖ Sélection des parents

La sélection est le lien entre les chromosomes et les résultats de leurs structures décodées, elle se fait aléatoirement en fonction de sa qualité d'adéquation liée à la qualité de résolution du problème. Cette opération permet aux chromosomes codants de bonnes structures de se reproduire le plus souvent que ceux qu'ils ne sont pas.

Plusieurs méthodes de sélection existent :

❖ Méthodes élitistes (N/2-élitisme)

Les individus sont triés selon leur fonction d'adaptation. Seule la moitié de la population, correspondante aux meilleurs composants, est sélectionnée. Nous pouvons constater que cette méthode induisait une convergence prématurée de l'algorithme : la pression de sélection est forte.

❖ Sélection par tournoi stochastique

Deux individus sont choisis au hasard et leur fonctions d'adaptation pour accéder à la génération intermédiaire est comparée, le plus adapté l'emporte avec une probabilité $P > 0.5$ (une valeur inférieure permet de réduire la pression de sélection si nécessaire). Cette étape est répétée jusqu'à ce que la génération intermédiaire, ce qui favorisera la pérennité de leur gènes.

❖ Sélection proportionnelle

C'est la méthode la plus utilisée (sélection par roue de fortune biaisée). Son principe est équivalent à une recherche linéaire sur une roue de fortune (loterie) dont les sections ont des tailles proportionnelles aux valeurs des fonctions d'aptitude des chaînes :

Soit F_i : la valeur de la fonction d'aptitude associée $i^{\text{ème}}$ individu.

Soit P_i : la probabilité pour que le $i^{\text{ème}}$ individu soit choisi pour participer à la recombinaison génétique :

$$P_i = \frac{F_i}{F_s} \quad (2-3)$$

$$C_i = \sum_{j=1}^n p_i \quad (2-4)$$

C_i La probabilité cumulée.

Le $i^{\text{ème}}$ individu est sélectionné si sa probabilité cumulée C_i est plus grande qu'une certaine valeur comprise entre 0 et 1 générée aléatoirement.

Une fois que l'opération de la sélection est terminée, les autres opérateurs entre en jeu, [7].

❖ Opérateur de croisement

Le croisement combine les gènes des deux individus parents pour donner deux nouveaux chromosomes d'individus enfants. En général, la zone de croisement est choisie aléatoirement dans les chromosomes. Les méthodes de croisement sont liées au codage.

Croisement binaire : croisement en un point (fig.2.8) s'effectue directement au niveau binaire.

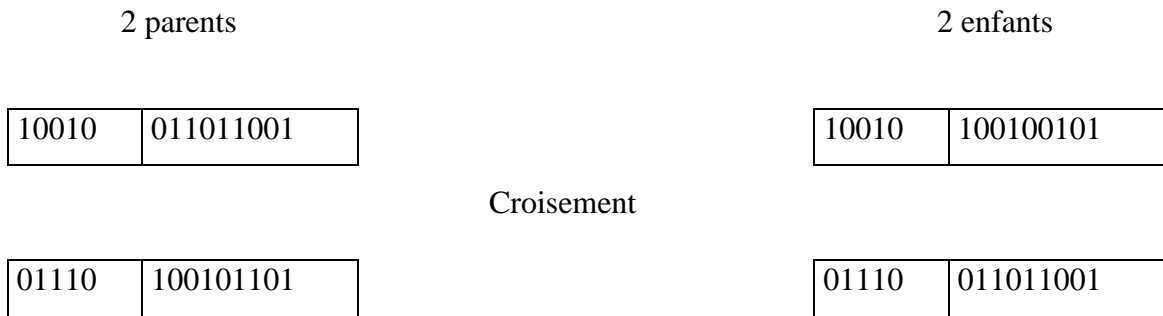
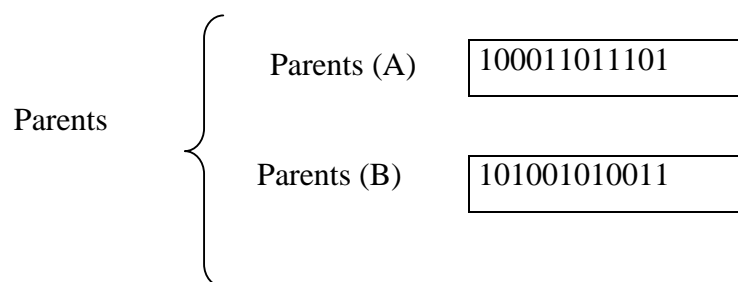


Figure. 2.7 Représentation schématique du croisement en 1 point

Il existe d'autres formes de croisement en K jusqu'au cas limite du croisement uniforme.
 $K=1,2,\dots, L-1$

Où L est la longueur de l'individu (nombre de bits de codage de chaque individu).

❖ **Croisement arithmétique** (figure 2.8) s'applique à une paire de chromosome et se résume à une moyenne pondérée des variables des deux parents.



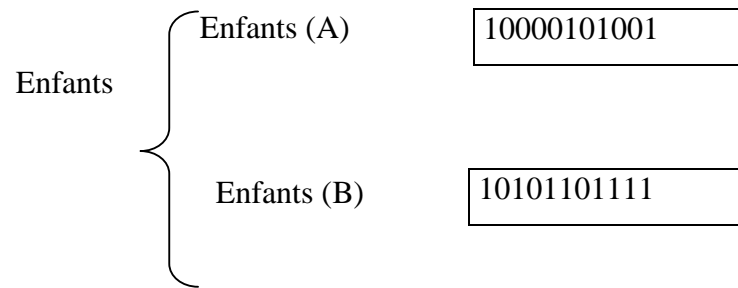
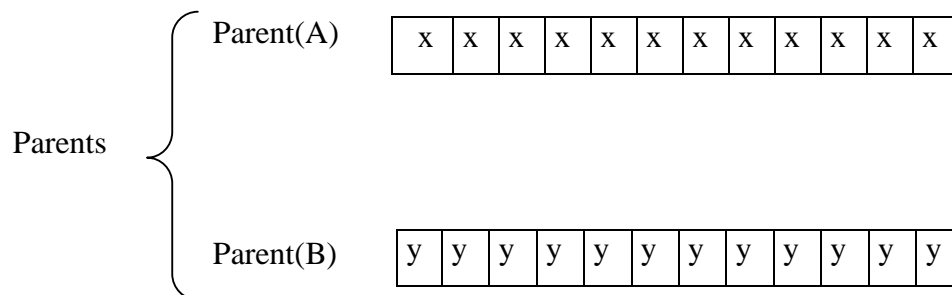


Figure. 2.8 Représentation schématique du croisement logique (ou arithmétique)

❖ **Croisement aléatoire** (figure 2.9) : à chaque position des nouveaux chromosomes, les caractères sont échangés en fonction d'une chaîne binaire aléatoire de même longueur des chromosomes qui est générée :

- Si la chaîne aléatoire contient un (1) à cette position, il y a changement.
- Si le bit aléatoire est un (0), aucun changement.



Chaîne aléatoire 0 1 0 1 1 1 0 0 1 0 0 1

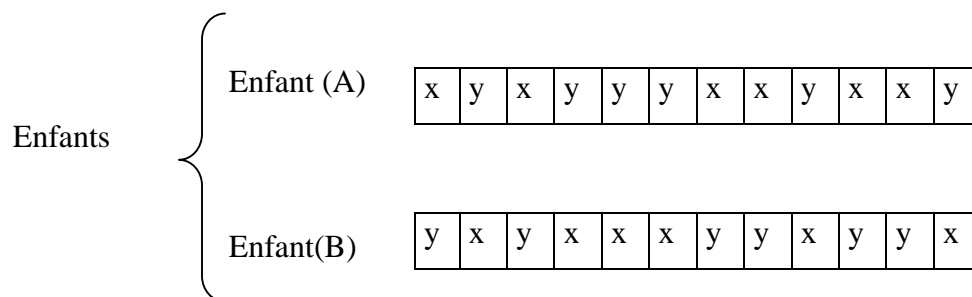


Figure. 2.9 Représentation schématique du croisement aléatoire

❖ **Croisement réel**

Le croisement réel ne se différencie du croisement binaire que par la nature des éléments qu'il altère : ce ne sont plus des bits qui sont échangés à droite du point de croisement, mais des variables réelles, [7].

❖ **Opérateur de mutation**

La mutation est une modification aléatoire d'un petit nombre de gènes, avec un faible taux de probabilité. Donc nous définissons une mutation comme étant l'inversion d'un bit dans un chromosome. Cela revient à modifier aléatoirement la valeur d'un paramètre du dispositif. Les mutations jouent le rôle de bruit et empêchent l'évolution de se figer, elles permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre des bits mutés. De plus, elles garantissent mathématiquement que l'optimum global peut être atteint.

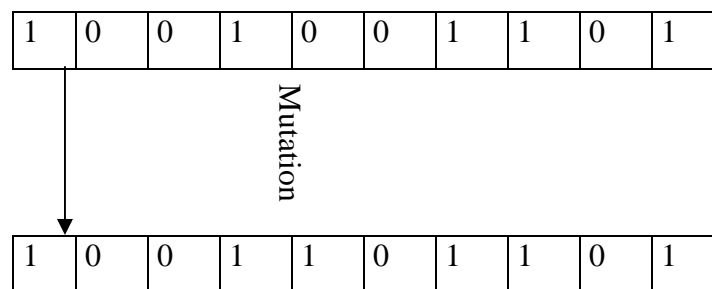


Figure. 2.10 Représentation schématique de la mutation

Comme pour le croisement, la mutation dépend du problème. On not :

❖ **Mutation avec le codage binaire :**

Un ou plusieurs gènes sont choisis aléatoirement, selon la taille du chromosome, et un taux très faible :

$$(\text{Taux} = \frac{\text{Nombre de gènes mutés}}{\text{Nombre de gènes totaux}})$$

Nombre de gènes totaux

❖ **Mutation avec le codage par valeur entières :**

Ici, le taux étant plus important (car un nombre plus grand de gènes sont mutés), il faut un taux de mutation plus faible que le taux dans le cas d'un codage binaire.

❖ **Mutation non uniforme :**

Possède la particularité de retirer les éléments qu'elle altère dans un intervalle de définition variable et de plus en plus petit, [7].

❖ Sélection des individus d'une nouvelle génération

A la suite de la recombinaison génétique (croisement et mutation), la population compte $2N$ individus (N parents et N enfants) ; N étant la taille de la population. Il faut donc éliminer N individus pour constituer la génération suivante. C'est le rôle de la sélection finale S_f qui agit sur la population des parents (\dot{P}_t) et des enfants (\ddot{P}_t) de la génération courante pour créer la génération suivante (P_{t+1}).

$$P_{t+1} = S_f(\dot{P}_t, \ddot{P}_t) \quad (2-5)$$

Pour effectuer cette sélection entre parents et enfants, plusieurs stratégies sont possibles

❖ Sélection par descendance :

La population de la nouvelle génération est obtenue par descendance, les enfants remplaçant automatiquement leurs parents quel que soit leur adaptation. Ce qui nous emmène vers la relation suivante :

$$P_{t+1} = S_f(\dot{P}_t, \ddot{P}_t) = \ddot{P}_t \quad (2-6)$$

L'inconvénient de ce mode de sélection est que l'on risque de voir disparaître les caractéristiques génétiques des parents les mieux adaptés si elles n'ont pas été totalement transmises lors de la recombinaison génétique.

❖ Sélection par compétition :

Une compétition a lieu entre parents et enfants pour déterminer les (survivants) de la génération. Ainsi, les enfants peuvent être insérés dans la population si et seulement si leur performance est supérieure à celle de leurs parents à rang équivalent.

$$a_{i+1} = \begin{cases} a''_i & \text{si } f(a''_i) > f(a'_i) \\ a'_i & \text{autrement} \end{cases} \quad (2-6)$$

Avec a_{i+1} est le $i^{\text{ème}}$ individu de la génération $(t+1)$, [7].

2.4.5 Convergence des algorithmes génétiques

❖ Critères de convergence

Le problème de convergence est évité en imposant un nombre maximale de génération t_{\max} et en arrêtant la recherche lorsque $t=t_{\max}$. On estime alors que l'algorithme

convergence et que l'individu de plus forte performance dans la population $P_{t_{max}}$ correspond à solution recherchée.

Une méthode plus rigoureuse consiste à supposer que l'algorithme converge vers l'optimum lorsque l'adaptation d'une partie de la population se rapproche de celle du meilleur individu. On peut considérer que cet événement se produit à la génération t pour laquelle

$$\frac{f_{\max} - f_{\text{moy}}}{f_{\max}} \leq \mu \quad (2-7)$$

Où : μ : La précision requise sur la convergence ;

f_{\max} : La performance du meilleur individu du population à la génération t ;

f_{moy} : La moyenne de l'adaptation calculée sur l'ensemble de la population ou sur une partie correspondant à un pourcentage des représentations les plus performantes.

❖ Taux de convergence

Du fait des problèmes de convergence et du caractère stochastique de l'exploration génétique, il est habituel d'exécuter plusieurs fois le même algorithme sur le même problème. Un taux de convergence (ou taux de réussite) rend compte de son efficacité. On peut par exemple définir le taux de convergence comme le rapport du nombre de fois où l'algorithme converge vers la solution optimale sur le nombre totale d'exécutions, [7].

2.4.6 Principaux paramètre d'un algorithme génétique

- ❖ La probabilité de croisement P_c .
- ❖ La probabilité de la mutation P_m .
- ❖ La taille de la population joue un rôle très important dans l'emploi des AG comme procédure d'optimisation. Si la taille est petit, cela implique une exécution rapide et une convergence incertaine. Si la taille est importante, le problème du temps de calcul apparaît.

2.5 Hybridation ou adaptation

Les AG dans leurs version de base (codage binaire, croisement en un point et mutation avec paramètres constants : P_c , P_m et la taille de population) sont des bonnes procédures d'optimisation, mais pas pour tous les types de problèmes.

Les difficultés d'emploi des AG comme méthode d'optimisation sont liées principalement au mode de représentation du problème sous une forme traitable par les AG

(Problème de codage). Donc, il est nécessaire d'adapter l'AG avec l'application désirée.

Une autre façon d'améliorer l'AG standard est hybridation avec d'autres méthodes classiques d'optimisation, ce qui permet d'exploiter l'expérience de la méthode classique et l'efficacité de l'AG dans un sens coopératif, [9].

2.5.1 Principe d'hybridation

Dais propose trois principes d'adaptation ou d'hybridation des AG avec d'autres environnements :

- ❖ Utiliser le codage courant du problème : utiliser si possible la même technique du codage de l'AG standard pour ne pas trop éloigner des principes de base ;
- ❖ Hybride là où c'est possible : incorporer les aspects positifs de la méthode classique dans l'algorithme hybride ;
- ❖ Adapter les opérateurs génétiques : créer des opérateurs bien adaptés au mode de codage utilisé par analogie avec les opérateurs génétiques employés dans l'AG standard (c'est-à-dire, incorporer les heuristiques propres au modèle sous forme de nouveaux opérateurs), [9].

2.5.2 Les approches d'hybrides

L'hybridation d'après Goldberg peut se faire selon deux approches :

- ❖ Approche séquentielle.
- ❖ Approche parallèle.

Dans l'approche séquentielle, l'AG effectue un nombre bien déterminé d'itération (de génération) et l'autre méthode avec laquelle s'hybride l'AG continue la recherche ou l'exploration d'une solution satisfaisante. Comme montre la figure suivante :

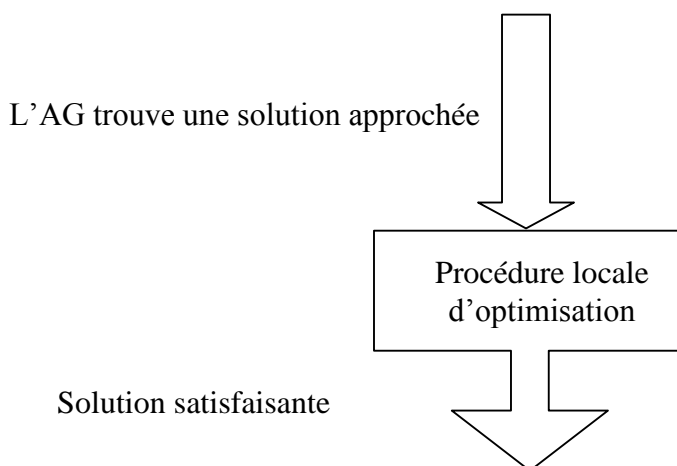


Figure. 2.11 L'approche parallèle séquentielle de l'hybridation

Dans l'approche parallèle, l'AG doit s'hybrider avec plusieurs processeurs en parallèle ayant une capacité de traitement suffisante pour que les évaluations des fonctions coût associées à chaque chaîne soient simultanément pour différentes chaînes dans une génération.

De cette façon ; les processeurs parallèles peuvent être utilisés pour l'évaluation des fonctions d'aptitude des chaînes, ils peuvent aussi être utilisés pour réaliser des fonctions du procédé d'exploration conventionnelle afin d'essayer d'améliorer la chaîne actuelle. La figure qui suit montre l'interaction de l'algorithme génétique (le maître) avec les processeurs (esclaves) (l'algorithme génétique dirige les activités des processeurs)

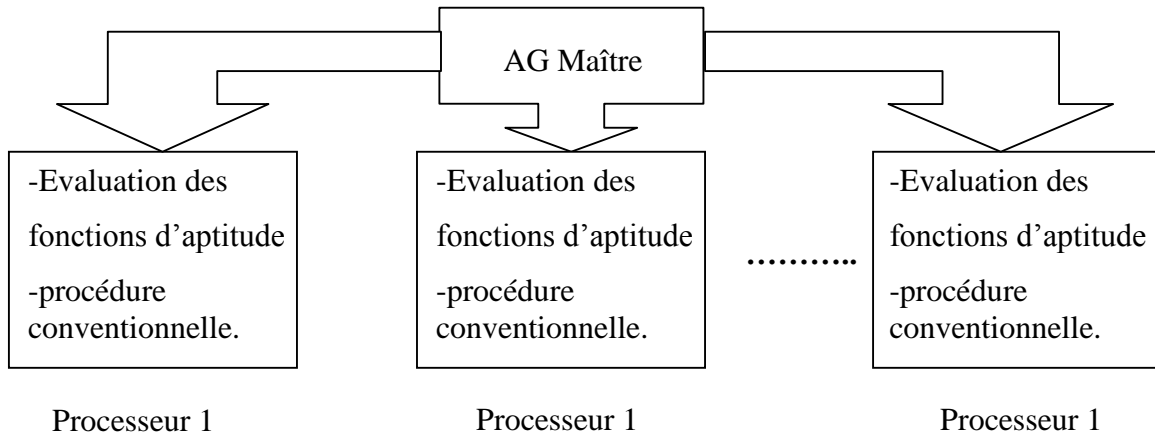


Figure. 2.12 L'approche parallèle de l'hybridation

Les techniques hybrides représentent une façon d'utiliser l'information auxiliaire et d'accélérer le processus d'exploration génétique d'un côté ; et elles permettent d'exploiter efficacement les connaissances sur la procédure conventionnelle, ce qui améliore la convergence d'un autre côté.

La notion d'exploitation des algorithmes génétiques sur des architectures parallèles a été introduite, mais les algorithmes génétiques n'ont pas fait l'objet de beaucoup de travaux dans la littérature du parallélisme. Les efforts théoriques et la mise en pratique ne font que commencer à recevoir une certaine attention, [9].

2.6 Avantages et inconvénients des algorithmes génétiques

Les points forts des algorithmes génétiques sont :

- ❖ Puissance de traitement d'une large gamme de types de données ;
- ❖ Adaptation aux problèmes d'optimisation ;
- ❖ Les résultats sont explicites et facile à analyser ;
- ❖ Convergence complète et rapide.

Les points faibles :

Les algorithmes génétiques sont des algorithmes :

- ❖ Stochastiques ;
- ❖ Ne renvoient aucune information sur l'optimalité de la solution trouvée ou sur la non faisabilité du problème ;
- ❖ Le codage binaire peut être complexe à réaliser l'optimalité, voire l'optimisation n'est pas garantie, [7].

2.7 Conclusion

Dans ce chapitre, nous avons présenté les bases fondamentales des algorithmes génétiques qui sont applicables à de nombreux problèmes, comme le problème du voyageur de commerce, et d'autres domaines.

Sur la base des connaissances précédentes, nous allons définir la structure de notre recherche, mise en œuvre dans le quatrième chapitre.