



DEUXIEME PARTIE

ALGORITHMES GENETIQUES

III.2.1 Introduction

L'idée à l'origine des algorithmes génétiques remonte aux années 50. Elle a émergé au travers des premières tentatives de modifier des programmes ou des paramètres avec du (pseudo) hasard, et d'évaluer le résultat, par Nils All Barricelli et Alex Fraser. [37]

Les algorithmes génétiques sont basés sur un phénomène naturel qui a fait ses preuves : *l'évolution*. Plus précisément, ils s'inspirent de l'évolution d'une population d'individus dans un milieu donné.

La théorie de ces algorithmes évolutionnaires est de simuler l'évolution d'une population d'individus divers (généralement tirée aléatoirement au départ) à laquelle on applique différents opérateurs (recombinaisons, mutations...) et que l'on soumet à une sélection, à chaque génération. Si la sélection s'opère à partir de la fonction d'adaptation, alors la population tend à s'améliorer

Un tel algorithme ne nécessite aucune connaissance du problème : on peut représenter celui-ci par une boîte noire comportant des entrées (les variables) et des sorties (les fonctions objectives). L'algorithme ne fait que manipuler les entrées, lire les sorties, manipuler à nouveau les entrées de façon à améliorer les sorties, etc. [34]

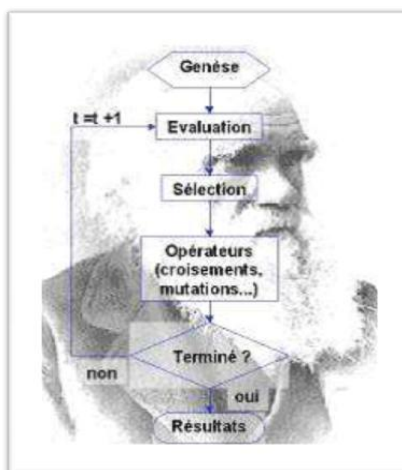


Figure III.10 organigramme des algorithmes évolutionnaires [34]



III. 2.2 objectif des algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'exploration développés à l'origine par J. Holland et son équipe au sein de l'université du Michigan [35] sont fondés sur la sélection naturelle et la génétique. Ils utilisent les principes de la survie des structures les mieux adaptées, les échanges d'informations pseudo-aléatoires. Ils reposent de manière intensive sur le hasard mais ne sont pas purement aléatoires.

III.2.3 Fonctionnement des algorithmes génétiques

Les AGs fonctionnent avec une population regroupant un ensemble d'individus (*chromosomes*). Pour chaque individu on attribue une valeur calculée par la fonction d'adaptation ou *fitness*.

En pratique, à partir d'une population, des chromosomes sont générés d'une façon aléatoire lors de l'initialisation. Pour définir la taille de la population, des travaux ont mentionné que cette taille varie d'un problème à un autre.

Dans chaque cycle d'opérations génétiques, une nouvelle population appelée *génération* est créée à partir des chromosomes de la population courante. Pour cela certains chromosomes appelés '*parents*' sont sélectionnés afin d'élaborer les opérations génétiques. Les gènes de ces parents sont mixés et recombinaés pour la production d'autres chromosomes appelés '*enfants*' constituant la nouvelle génération.

Les étapes de l'AG sont répétées durant t cycles, l'arrêt de l'algorithme est fixé d'après un *critère d'arrêt*. On peut avoir plusieurs critères d'arrêt :

- ✓Le nombre de génération fixé initialement a été atteint.
- ✓La valeur de la fonction d'adaptation a atteint une valeur fixée *a priori*.
- ✓L'absence d'évolution de la valeur de la fonction d'adaptation des individus d'une population à une autre.
- ✓Les chromosomes ont atteint un certain degré d'homogénéité.

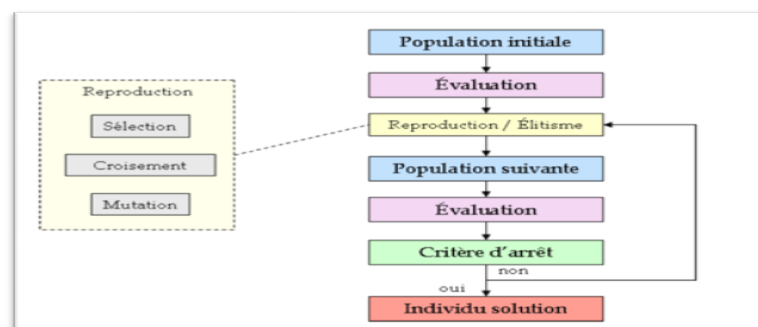


Figure III.11 schéma de principe d'un algorithme génétique



Globalement l'algorithme est basé sur :

- Une représentation chromosomique des solutions du problème.
- Une méthode pour créer une population initiale de solutions.
- Une fonction d'évaluation (fitness) pour classer les solutions en fonction de leurs aptitudes.
- Des opérateurs génétiques qui définissent la manière dont les caractéristiques génétiques des parents sont transmis aux descendants (enfants).
- Les valeurs des paramètres utilisés par l'AG.

III.2.4 Codage et opérateurs d'un algorithme génétique

Trois opérateurs caractérisent les algorithmes génétiques et rappellent l'origine de ces méthodes, ils vont permettre à la population d'évoluer, par la création d'individus nouveaux construits à l'aide des individus anciens. Plus précisément, on prélève, dans certains individus de la population courante, une partie de leurs caractéristiques en choisissant certaines parties des chromosomes qui les représentent ; puis on recombine ces différentes parties pour constituer les individus de la nouvelle population :

La phase de sélection indique dans quelles configurations de la population courante on va prélever des morceaux de chromosomes ; la phase de croisement prélève ces morceaux de chromosomes et les recombine pour former les configurations de la population suivante.

La phase de mutation s'applique à la nouvelle population en changeant éventuellement certains gènes de certains chromosomes obtenus à la fin de la phase de croisement. Une succession des trois opérations de sélection, de croisement et de mutation constitue une génération, et les algorithmes génétiques consistent donc à faire évoluer une population initiale pendant un certain nombre de générations, nombre déterminé par l'utilisateur. [38]

III.2.4.1 : le codage

Une méthode évolutionnaire, spécifiquement basée sur un algorithme génétique nécessite une représentation des individus qui sont les solutions du problème que l'on cherche à résoudre. Dans la plupart des représentations chaque gène correspond à un attribut qui représente une variable de l'espace de recherche. Chaque variable, associée à une caractéristique de l'espace peut prendre certaines valeurs (discrètes ou continues). Une représentation simple, souvent utilisée en algorithme génétique, adapte le principe de gènes binaire exprimant la présence



ou l'absence de la caractéristique associée .dans de telle représentation, chaque individu est représenté par une chaîne de longueur généralement fixe

Aujourd'hui, il est unanimement admis qu'il faut considérer 2 espaces : l'espace où est posé le problème, sur lequel on peut calculer la fonction objectif ou espace phénotypique, et l'espace où travaillent les opérateurs génétiques ou espace génotypique

Le codage est le passage du phénotype ou génotype, un bon codage est celui qui favorise la fonction des briques élémentaires qui seront recombinaisonnée par l'opérateur de croisement.

Il y a deux principaux types de codage utilisables, et on peut passer de l'un à l'autre relativement facilement :

✓**Codage binaire** : Chaque gène dispose du même alphabet binaire {0, 1}, C'est le plus utilisé et celui qui a été employé lors de la première application des algorithmes génétiques.

Chromosome A

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

Chromosome B

1	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---

Figure III.12 Exemple de codage binaire

✓**Le codage par permutations** de valeurs entières : Le gène est codé par une valeur entière dans un ensemble de cardinalité égale au nombre de gènes, cela peut-être utile notamment dans le cas où l'on recherche le maximum d'une fonction réelle.

Exemples :

Chromosome A

1	3	6	2	5	8	4	7
---	---	---	---	---	---	---	---

Chromosome B

6	1	3	2	7	4	5	8
---	---	---	---	---	---	---	---

Figure III.13 Exemple de codage par permutation

III.2.4.2 : Initialisation de la population

cette étape consiste à initialiser une population avec une méthode complètement aléatoire, la population se compose d'un ensemble d'individus (points dans l'espace de recherche),solution possible du problème.



Un individu, appelé généralement chromosomes est constitué de gènes de chaînes de bits codés souvent en binaire '0-1', à chaque individu on attribue une fonction performance évaluant le mérite de cet individu en tant que solution possible du problème la population évolue en une succession de générations en respectant le principe que les individus les plus adaptés (en terme de valeurs de la fonction performance) survivent et se reproduisent

III.2.4.3 : Principe de sélection

Cet opérateur est chargé de définir quels seront les individus de P qui vont être dupliqués dans la nouvelle population P' et vont servir de parents (application de l'opérateur de croisement). Soit n le nombre d'individus de P, on doit en sélectionner $n/2$ (l'opérateur de croisement nous permet de repasser à n individus).

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population.

Il existe différentes méthodes de sélection :

- La méthode de la "loterie biaisée" (roulette wheel) de Goldberg,
- La sélection par tournois,

La loterie biaisée ou roulette Wheel

Cette méthode est la plus connue et la plus utilisée. Avec cette méthode chaque individu a une chance d'être sélectionné proportionnelle à sa performance, donc plus les individus sont adaptés au problème, plus ils ont de chances d'être sélectionnés.

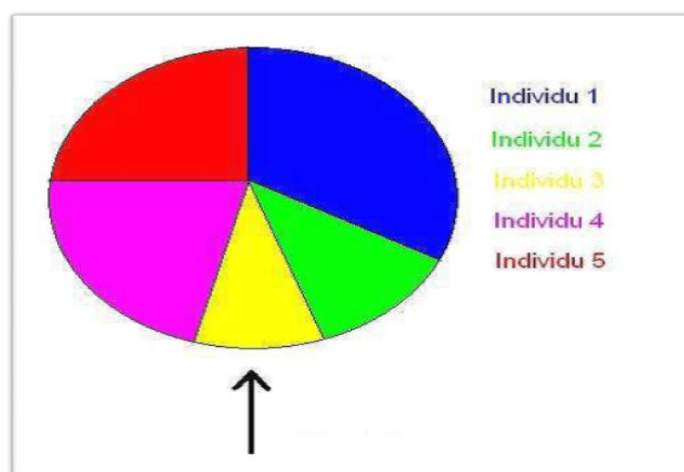


Figure III.14 Méthode de sélection de la loterie biaisée[35]



La sélection par tournois

Cette méthode est celle avec laquelle on obtient les résultats les plus satisfaisants.

Le principe de cette méthode est le suivant : on effectue un tirage avec remise de deux individus de P , et on le fait "combattre". Celui qui a la fitness la plus élevée l'emporte avec une probabilité p comprise entre 0.5 et 1. On répète ce processus n fois de manière à obtenir les n individus de P' qui serviront de parents.

La variance de cette méthode est élevée et le fait d'augmenter ou de diminuer la valeur de p permet respectivement de diminuer ou d'augmenter la pression de la sélection.

III.2.4.4 : principe de croisement (crossover)

Le crossover utilisé par les algorithmes génétiques est la transposition informatique du mécanisme qui permet, dans la nature, la production de chromosomes qui héritent partiellement des caractéristiques des parents.

Son rôle fondamental est de permettre la *recombinaison* des informations présentes dans le patrimoine génétique de la population.

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Ils consistent à échanger les gènes des parents afin de donner des enfants qui portent des propriétés combinées. Bien qu'il soit aléatoire, cet échange d'informatique offre aux algorithmes génétiques une part de leurs puissances : quelque fois de bons gènes d'un parent viennent les mauvais gènes d'un autre et créent des fils mieux adaptés aux parents.

Il existe différentes techniques de croisement

Croisement à 1 point (croisement simple).

On choisit aléatoirement un point de croisement pour chaque couple d'individus sélectionnés. Notons que le croisement s'effectue directement au niveau des gènes représentés soit en binaires ou en réels. Un chromosome ne peut pas être coupé au milieu d'un gène. La figure II.15 illustre ce croisement d'un seul point de coupure dans le cadre d'une représentation binaire ou réelle des gènes des chromosomes.

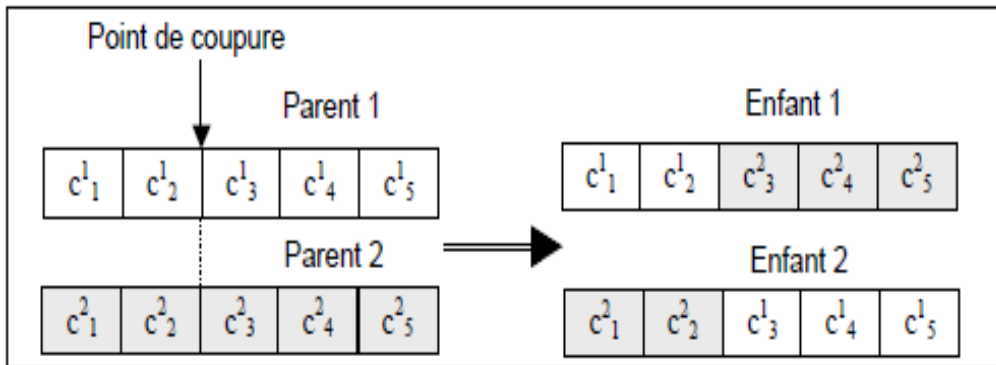


Figure III.15 : Croisement à 1 point

a. Croisement multiple (multipoint)

Plusieurs auteurs se sont penchés sur l'utilisation de plusieurs points de coupure concernant l'opérateur de croisement. Le nombre de points de coupure généré est en moyenne $L/2$ où L est la taille du chromosome.

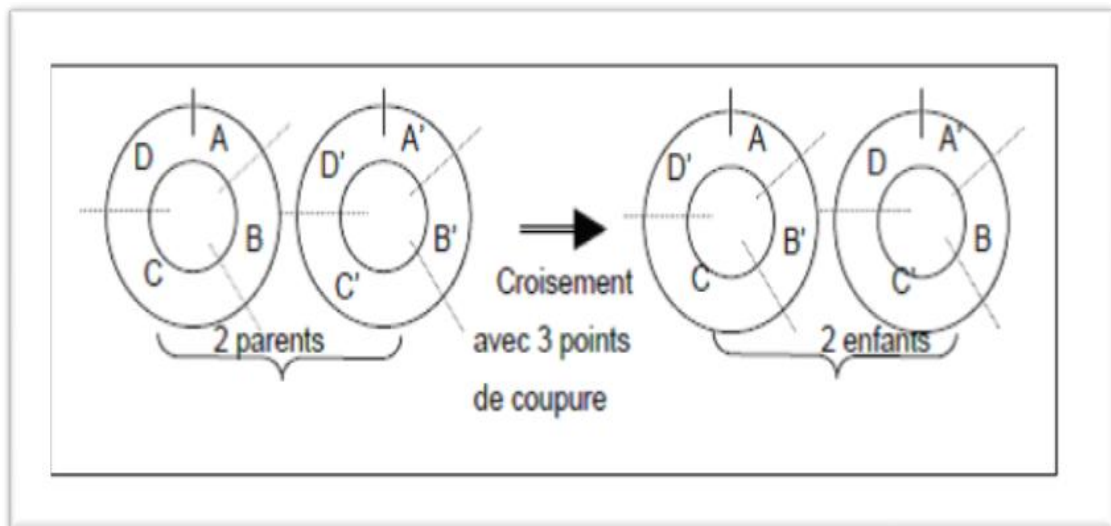


Figure III.16- Croisement avec 3 points de coupure

III.2.4.5 : principe de l'opérateur de mutation

Cet opérateur consiste à changer la valeur allélique d'un gène avec une probabilité pm très faible, généralement comprise entre 0.01 et 0.001.

On peut aussi prendre $pm = 1 / lg$ où lg est la longueur de la chaîne de bits codant notre chromosome.

Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits, mais vu la



probabilité de mutation c'est extrêmement rare) se trouvant en un locus bien particulier et lui aussi déterminé de manière aléatoire; on peut donc résumer la mutation de la façon suivante :

On utilise une fonction censée nous retourner *true* avec une probabilité *pm*.

Pour chaque locus **faire**

Faire appel à la fonction

Si cette fonction nous renvoie *true* **alors** on inverse le bit se trouvant à ce locus

FinSi

FinPour

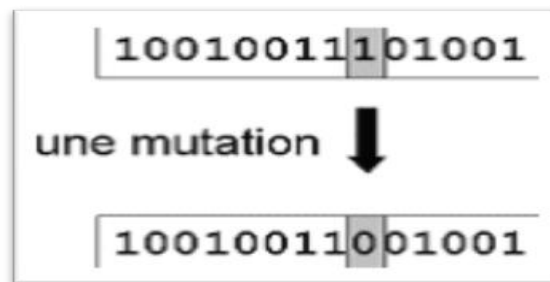


Figure III.17 - une mutation

L'opérateur de mutation modifie donc de manière complètement aléatoire les caractéristiques d'une solution, ce qui permet d'introduire et de maintenir la diversité au sein de notre population de solutions. Cet opérateur joue le rôle d'un "élément perturbateur", il introduit du "bruit" au sein de la population.

III.2.5 Avantages et inconvénients des algorithmes génétiques

L'avantage principale des AGs est la recherche de l'optimum global par contre ils consomment beaucoup de temps d'exécution



TROISIEME PARTIE

HYBRIDATION ALGORITHMES GENETIQUES

& RESEAUX DE NEURONES

III.3.1 Introduction

Les réseaux évolutionnaires sont des réseaux neuronaux qui utilisent des algorithmes génétiques avec au moins trois finalités principales :

- ✓-l'évolution des poids des connexions neuronales
- ✓-l'évolution de l'architecture du réseau, l'idée étant de remplacer les procédures manuelles d'essais et d'erreurs par un algorithme évolutionnaire ;
- ✓L'évolution des règles d'apprentissage .On cherche alors la règle la mieux adaptée au problème.

Dans la pratique, il est naturellement possible de combiner ces différentes approches .l'algorithme suit typiquement les étapes suivantes :

- 1) construction d'une population de génomes représentant des réseaux.
- 2) décodage des génomes et génération des réseaux
- 3) calcul de la fitness de chacun des réseaux .la fonction fitness est basée sur l'erreur total, la durée de convergence, la complexité du réseau, ect.
- 4) si l'objectif (ou le nombre maximum d'itérations)est atteint sortie de l'algorithme
- 5) sélection des individus les meilleurs pour reproduction
- 6) application des différentes opérateurs génétiques pour construire les génomes des descendants
- 7) sélection de la nouvelle génération parmi l'ensemble formé par les populations initiales et descendants et remplacement de la population
- 8) retour à l'étape 2

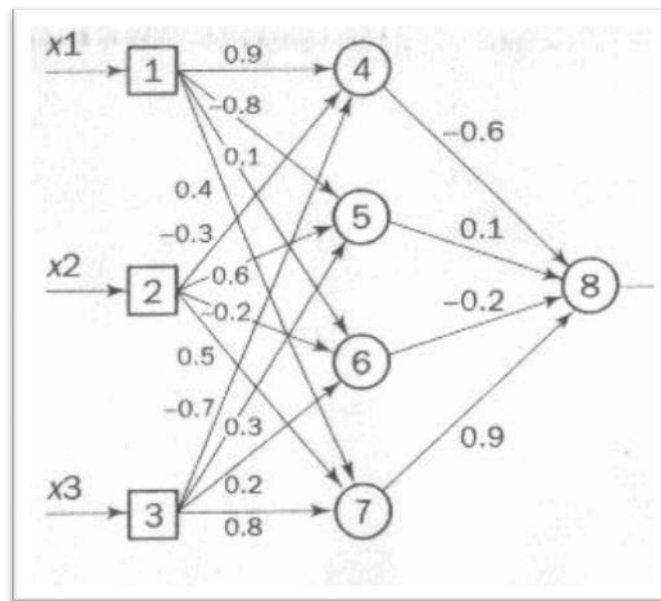
III.3.2 Utilisation des AGS pour une optimisation des poids

(Apprentissage paramétrique)

La partie suivante présente le concept de base d'une technique d'optimisation de poids génétiques [39], [40], [41]



Pour une utilisation des algorithmes génétiques, il faut d'abord représenter le problème comme un chromosome. Par exemple, nous voulons optimiser les poids d'un perceptron multicouche présenté dans la figure III.18



Chromosome:

0.9	-0.3	-0.7	-0.8	0.6	0.3	0.1	-0.2	0.2	0.4	0.5	0.8	-0.6	0.1	-0.2	0.9
-----	------	------	------	-----	-----	-----	------	-----	-----	-----	-----	------	-----	------	-----

Figure III.18 - présentation d'un système neuro-génétique [39]

Dans **la première étape** on va générer Des poids initiaux dans le réseau choisi aléatoirement dans le petit intervalle [-1,1]. Dans ce perceptron, il y a 16 liaisons pondérées entre les neurones. Puisqu'un chromosome est un ensemble de gènes, l'ensemble des poids peut être représenté par un chromosome à 16 gènes, où chaque gène correspond à une liaison simple pondérée dans le réseau. Ce chromosome présente un individu d'une population c.ad une solution proposé à partir d'un ensemble des solutions

Dans **La deuxième étape** on doit définir une fonction d'évaluation (fitness) pour évaluer la performance des chromosomes. Cette fonction doit estimer la performance d'un réseau neuronal donné. Nous pouvons appliquer ici une fonction assez simple définie par la réciproque de l'erreur quadratique

On peut utiliser aussi comme une fonction le taux de classification non correcte

La **troisième étape** on doit appliquer les deux opérateurs des algorithmes génétiques croisement et mutation. Figure III.19 et III.20

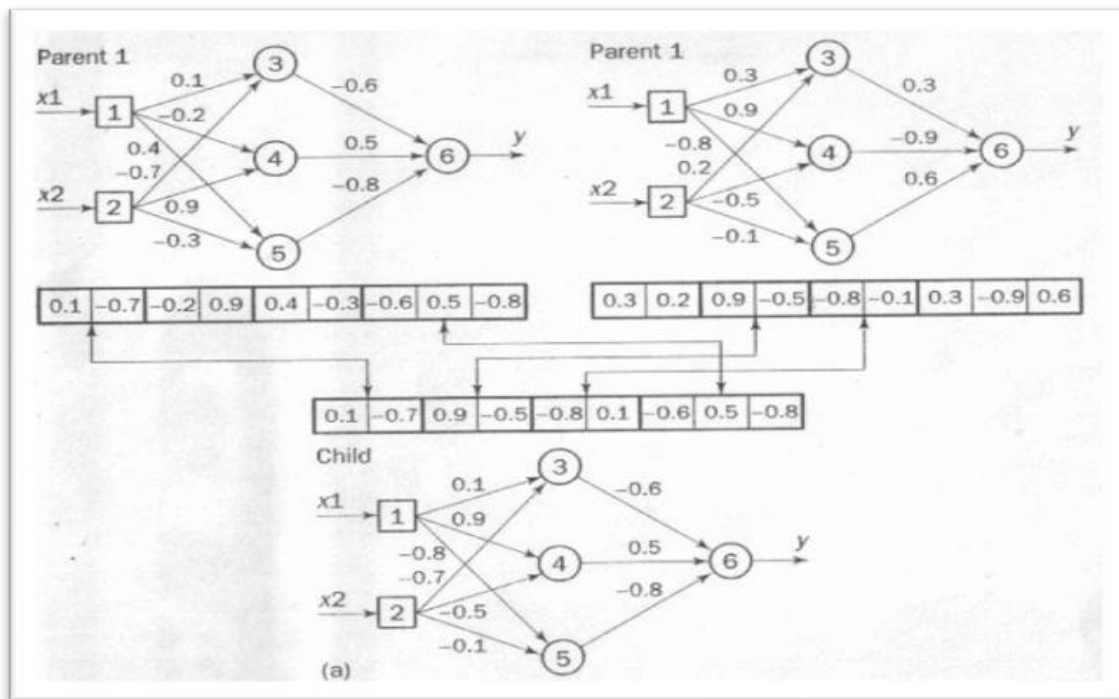


Figure III.19 Opérateur de croisement dans un système neuro-génétique [39]

Un opérateur de mutation choisit aléatoirement un gène dans un chromosome et ajoute une petite valeur aléatoire à chaque poids dans ce gène..

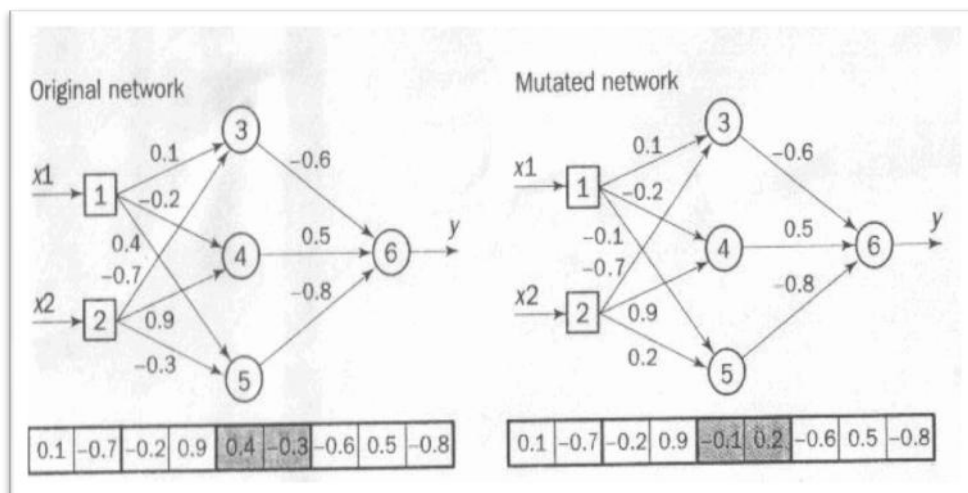


Figure III.20 - opérateur de mutation dans le système neuro-génétique [39]

Maintenant nous sommes prêts à appliquer l'algorithme génétique. Bien sûr, nous devons toujours définir la taille de population, c'est-à-dire le nombre de réseaux avec des poids différents, la probabilité de croisement et de mutation et le nombre de générations



Evolution de l'architecture des réseaux :

L'évolution de l'architecture des réseaux associe à un algorithme génétique qui engendre une population de réseaux et un processus d'apprentissage qui permet d'en évaluer les fitness. On utilise des fonctions complexes pour les architectures de réseaux. Outre l'erreur totale ou la vitesse de convergence, on doit en effet tenir compte de la complexité du réseau (nombre de connexions et de neurones)

La méthode consiste à encoder l'ensemble des caractéristiques du réseau. On peut alors utiliser pour un réseau de N neurones, une matrice carrée de N*N dont les valeurs booléennes spécifient l'(in)existence d'une connexion .le chromosome binaire en découle directement[36]

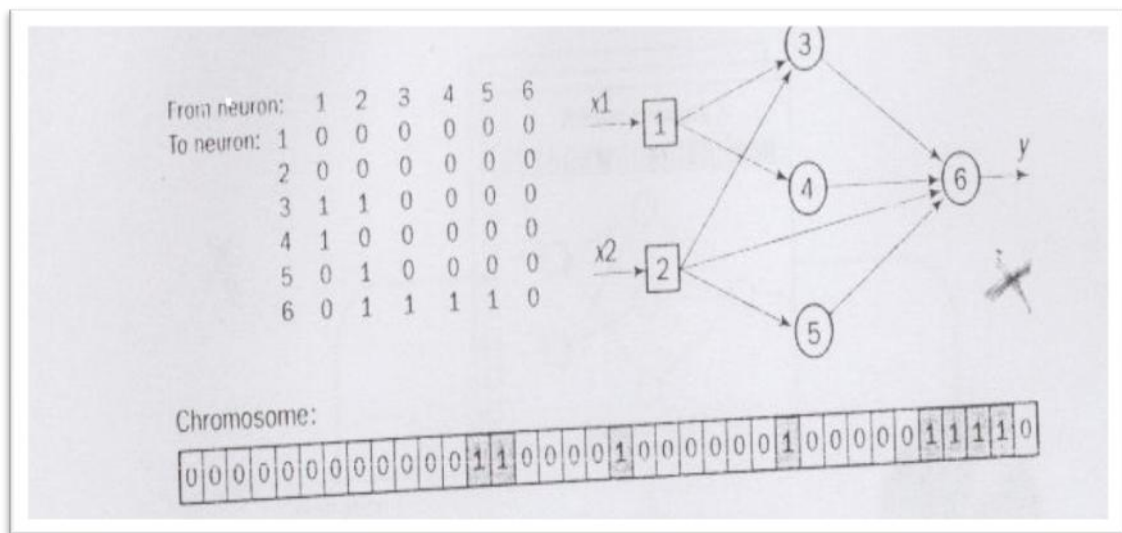


Figure III.21- Système neuro-génétique pour le choix de la topologie [36]

Il est naturellement possible de substituer à la matrice booléenne une matrice réelle .on peut traiter à la fois de l'architecture et de l'apprentissage. [34]

III.3.3 Conclusion

Les RNS peuvent modifier les poids synaptiques et leurs apprentissages classique (rétro-propagation)ou un apprentissage hybride en faisant appel aux AGS.

Dans ce travail nous avons adopté les deux approches