

Matière : Intelligence artificielle- 3SI

Cours 6 Les Algorithmes génétiques

Introduction

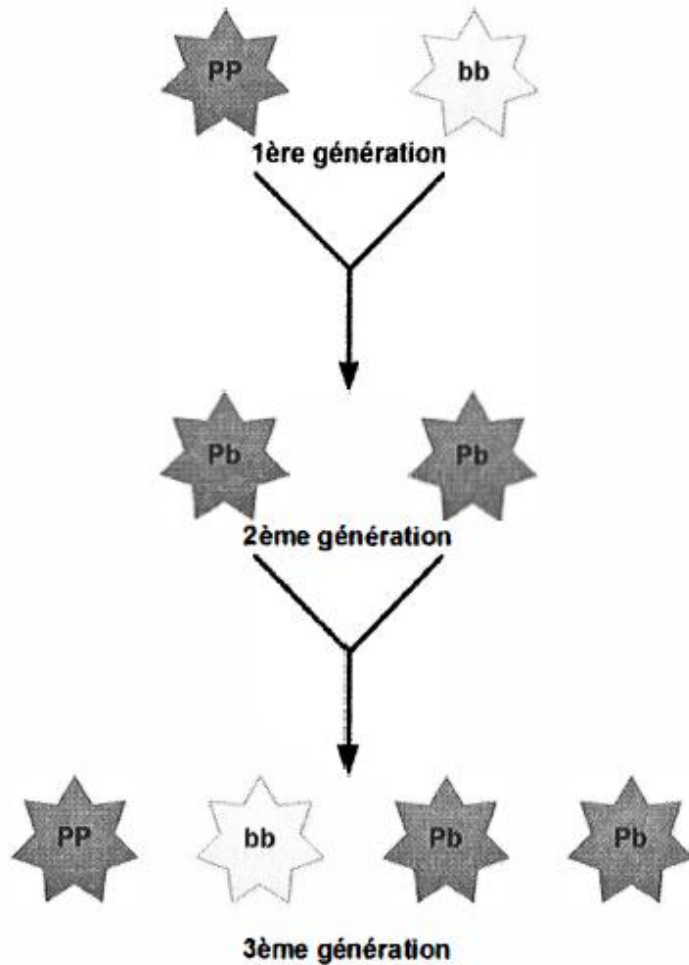
- Les algorithmes génétiques sont basés sur **l'évolution biologique**.
- L'évolution biologique fut étudiée à partir de la fin du 18e siècle.
- De plus, les sociétés **d'élevage** : sélectionner **les meilleurs** individus d'un animal pour **améliorer la production** d'une espèce (comme le lait pour les vaches). Les races de chiens, de chats ou de chevaux **étaient ainsi nombreuses et fort différentes**. Il était évident qu'un animal était **proche de ses parents**, bien que pas totalement identique à ceux-ci, et qu'en sélectionnant les **bons parents**, on pouvait créer de **nouvelles races**.

2. Les causes des mutations

Darwin (1809 - 1882) et Lamarck (1744 - 1829) s'opposèrent sur les raisons des **modifications** entre les parents et les descendants, appelées **mutations**.

D'après Darwin, **un descendant était la moyenne de ses parents**, mais parfois, aléatoirement, des différences apparaissaient . Seuls les individus **les plus adaptés pouvant survivre** et donc se **reproduire**, seules **leurs modifications se transmettaient à leurs descendants**. Les mutations qui n'étaient pas intéressantes au sens de la survie n'étaient donc pas propagées, et s'éteignaient.

Au contraire, celles qui apportaient un avantage sélectif étaient conservées et se propageaient.



On voit aussi ici pourquoi on dit que certaines caractéristiques "sautent u ne génération" (on obtient **une fleur blanche à la troisième** génération bien qu'on n'en ait pas à la deuxième). Il s'agit en fait de **traits récessifs**.

3. Des facteurs au code génétique

L'ADN fut isolé en 1869, puis les chromosomes en 1879 par Flemming.

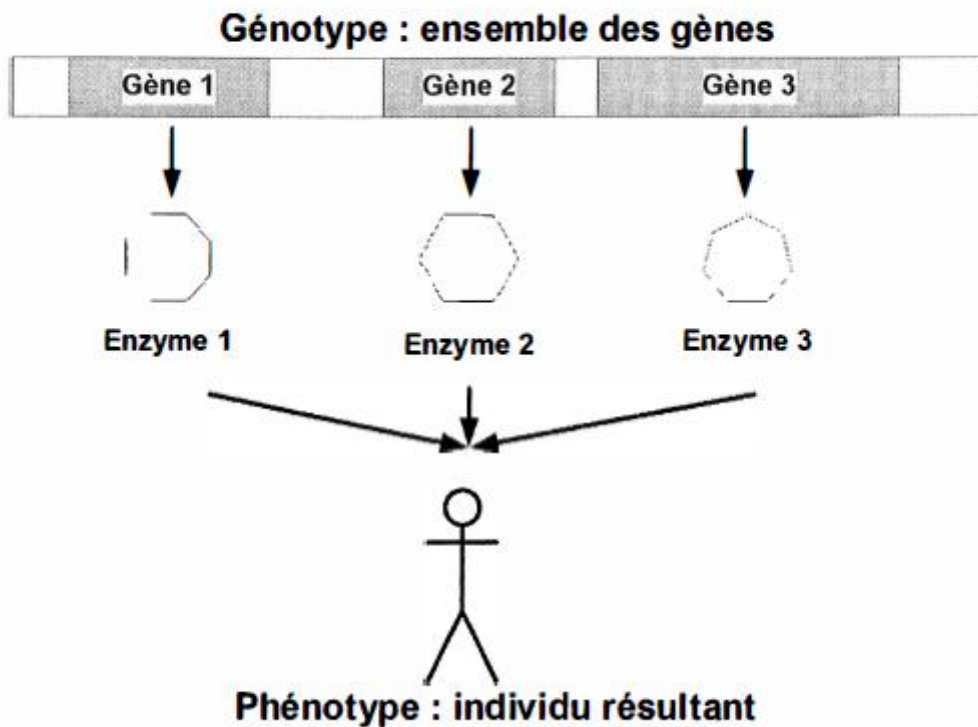
Dans l'ADN, des chromosomes se situaient les **facteurs de Mendel**.

On parle dès 1909 de **gènes** au lieu de facteurs, et une première carte des gènes (son chromosome X) fut d'ailleurs proposée dès 1913.

La structure de l'ADN en double fut découverte en 1952 par Watson, Crick et Wilkins. **Le code génétique** fut admis dans les années 1960.

Les principes basiques de la génétique sont alors tous présents : un individu possède des **chromosomes**, contenant des **gènes**. Chaque gène correspond à **une enzyme**, grâce à un code qui indique la composition de celle-ci. L'ensemble des gènes d'un être vivant est appelé son **génotype**. Les interactions entre toutes les enzymes créent l'individu, appelé **phénotype**.

Voici un schéma récapitulatif :

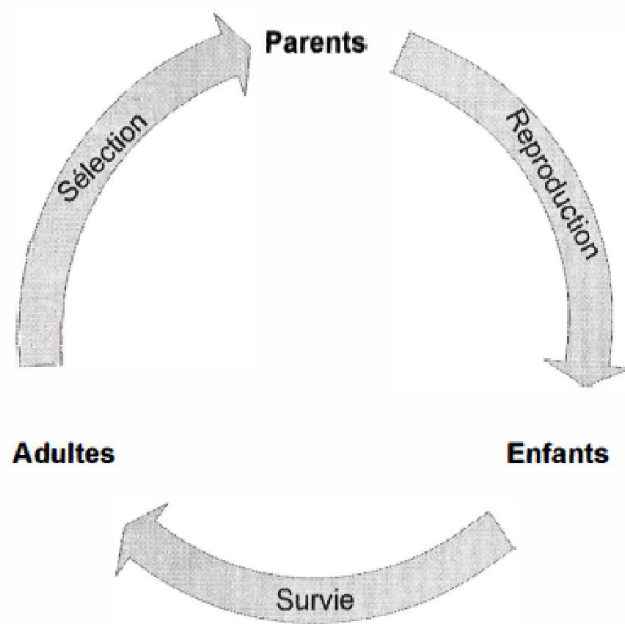


En résumé, un individu possède un ensemble de gènes (le génotype) , présents à chaque fois en double exemplaire. La transcription puis la traduction permettent de transformer ces gènes en enzymes, qui vont réagir entre elles et créer l'être vivant (le phénotype) .

Lors de la reproduction, il va donner à son descendant la moitié de son capital génétique, qui sera mixé avec le capital génétique du deuxième parent. De plus, pendant ce processus, des mutations aléatoires peuvent se produire.

La sélection naturelle va donc avantager les mutations et les croisements d'individus intéressants pour la survie de l'espèce. Ceux-ci vont se disséminer dans la population et l'espèce va continuellement s'améliorer et s'adapter à son environnement .

On peut résumer ce "cercle de la vie" par la figure suivante :



4. Évolution artificielle

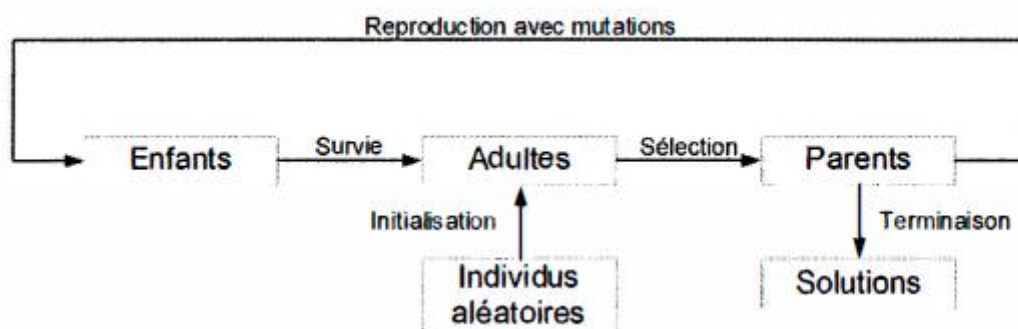
4. 1 Principes

Les algorithmes évolutionnaires vont donc partir d'une population de solutions potentielles à un problème. Chacune est évaluée, pour lui attribuer une note, appelée fitness. Plus la fitness d'une solution est forte et plus celle-ci est prometteuse.

Les meilleurs individus sont ensuite sélectionnés, et se reproduisent. Deux opérateurs artificiels sont alors utilisés : le croisement entre deux individus, appelé **crossover**, et des mutations aléatoires.

Une étape de survie s'applique alors pour créer la nouvelle génération d'individus.

Le processus est donc le suivant :



4.2 Vue d'ensemble du cycle

4.2. 1 Phases d'initialisation et de terminaison

Lors de la phase d'initialisation, une première population est créée. Pour la plupart des problèmes, on part de solutions aléatoires, qui sont donc en moyenne très peu adaptées au problème. Si on connaît déjà des solutions acceptables au problème, il est possible de directement les injecter lors de l'initialisation. Le processus complet est alors plus rapide, nécessitant moins de générations.

Il faut aussi définir un critère d'arrêt . Celui-ci permet de savoir à quel moment s'arrêter, pour donner à l'utilisateur les meilleures solutions trouvées. Ce critère peut porter sur **un nombre de générations ou sur une fitness minimale à obtenir par les individus (un "score" à atteindre)**

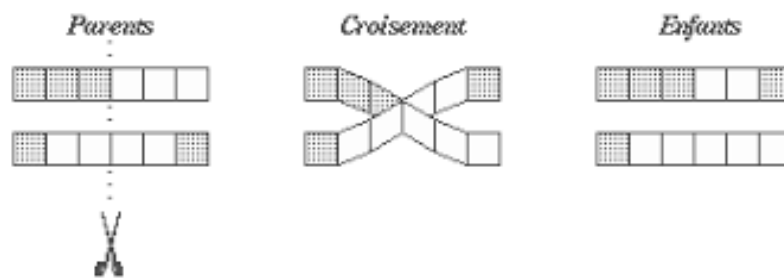
4.2.2 Phase de sélection

La sélection consiste à **déterminer quels sont les individus qui méritent d'être choisis comme parents pour la génération suivante.**

Parfois en faisant muter ou en croisant des solutions en apparence "mauvaises" que l'on peut trouver une bonne solution à un problème.

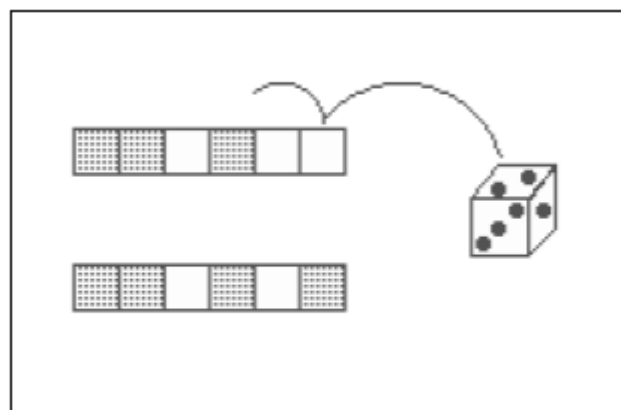
4.2. 3 Phase de reproduction avec mutations

Lors de la reproduction, on choisit pour chaque enfant de un à N parents. Les informations génétiques des différents parents sont mixées avec l'opérateur de **crossover**.



Croisement

Une fois le mélange des parents effectué, on applique des mutations choisies aléatoirement au résultat, et dont le nombre dépend du taux de mutation de l'algorithme .



Mutation

4.2.4 Phase de survie

Les enfants étant créés, il faut maintenant obtenir **une nouvelle génération d'adultes** qui peuvent ou non se reproduire. Si la solution la plus simple consiste à remplacer toute la génération des parents par la génération des enfants, il existe cependant plusieurs autres stratégies de survie . On obtient donc à la fin de la survie une nouvelle population, et on peut reboucler tout le processus.

4.3 Convergence

La convergence vers la solution optimale est démontrée théoriquement. Cependant, rien ne précise le temps nécessaire pour converger vers cette solution, qui peut donc être supérieur à ce qui est acceptable .

Il est donc important de bien choisir les différents opérateurs (sélection, mutation, crossover et survie) et les représentations, au nombre de trois : des gènes, des individus et de la population.

EXEMPLE :

Un algorithme génétique va reproduire le modèle d'évolution dans le but de trouver des solutions pour un problème donné:

- dans notre cas, une population sera un ensemble d'individus ;
- un individu sera une solution à un problème donné ;
- un gène sera une partie d'une solution, donc d'un individu ;
- une génération est une itération dans un algorithme.

Le déroulement d'un algorithme génétique peut être découpé en cinq parties :

1. La création de la population initiale ;
2. L'évaluation des individus ;
3. La création de nouveaux individus ;
4. L'insertion des nouveaux individus dans la population ;
5. Répétition du processus.

Pour illustrer , on montrera comment appliquer un algorithme génétique au problème bien connu de « l'informaticien en vacances . L'informaticien en vacances doit visiter plusieurs villes durant ses vacances : { A,B,C,D,E,F,G,H,I,J } . **Il cherche donc le chemin le plus court pour passer par chacune d'elles. Il souhaite également assister à un maximum de festivals musicaux (ayant lieu dans les villes A,B et C). Il s'agit donc d'un problème bicritère (la distance entre les villes à minimiser et le nombre de festivals auxquels il pourra assister à maximiser).** Chaque festival a lieu à une date donnée. Nous connaissons aussi les distances entre toutes les villes.

I. La création de la population initiale

Pour démarrer un algorithme génétique, il faut lui fournir une population à faire évoluer. La manière dont le programmeur va créer chacun des individus de cette population est entièrement libre. Il suffit que tous les individus créés soient **de la forme d'une solution potentielle**, et il n'est nullement besoin de songer à créer de bons individus.

Il est tout à fait possible de créer les **individus de manière aléatoire**. Et cette méthode amène un concept très utile dans les algorithmes génétiques : **la diversité**. Plus les individus de la population de départ seront différents les uns des autres, plus nous aurons de chances d'y trouver, non pas la solution parfaite, mais de quoi fabriquer les meilleures solutions possibles.

Problème de l'informaticien en vacances
Nous n'allons pas parcourir tous les chemins possibles, il y en aurait trop. Nous créons une population d'individus au hasard : par exemple {A,B,C,D,E,F,G,H,I,J}, {D,A,F,J,C,E,G,H,B,I}, {J,A,H,F,D,C,I,G,E,B} ou encore {J,F,A,C,B,E,I,H,G,D}. Une population intéressante pourrait être de taille 100, l'essentiel étant que les individus contiennent toutes les villes une et une seule fois, et qu'ils soient relativement différents les uns des autres.

La taille de la population à manipuler

La taille de la population initiale est également laissée à l'appréciation du programmeur.

- pour la qualité des solutions trouvées
- pour le temps d'exécution de l'algorithme.

L'évaluation des individus

Une fois que la population initiale a été créée, nous allons en sortir les individus les plus prometteurs, ceux qui vont participer à l'amélioration de notre population. Nous allons donc attribuer une '**note**' ou un **indice de qualité** à chacun de nos individus. La méthode d'évaluation des individus est laissée au programmeur en fonction du problème qu'il a à optimiser ou à résoudre.

Méthodes d'évaluation d'individus en contexte multicritère

Les méthodes utilisent la notion de **dominance d'individu**.

On dit qu'un individu en domine un autre s'il est meilleur **dans chacun des critères**, pris séparément.

Problème de l'informaticien en vacances
L'évaluation du critère relatif à la distance parcourue est simplement la somme de toutes les distances parcourues. Et l'évaluation du critère relatif au nombre de festivals est simplement le nombre de festivals auxquels l'informaticien a pu assister (0,1,2 ou 3). Une solution amenant {50 km, 2 festivals} dominera une solution {60 km, 1 festival}, mais ne dominera pas une solution {40 km, 3 festivals}, car elle n'est pas meilleure sur tous les critères.

L'étape d'évaluation des individus peut être effectuée avant et/ou après les étapes de croisement et de mutations.

La création de nouveaux individus

A. La sélection

Nous allons maintenant **enrichir notre population en croisant des individus**. Nous allons essayer de prendre des **morceaux de solution de certains individus et d'autres morceaux d'autres individus pour créer des nouveaux individus** qui, on l'espère, seront des solutions meilleures à notre problème.

Il est tout à fait possible de choisir des individus au hasard et de les mélanger aléatoirement pour créer de nouveaux individus.

A-1. La roulette

La sélection des individus par le système de la roulette s'inspire des roues de loterie. À chacun des individus de la population est associé un secteur d'une roue. L'angle du secteur étant proportionnel à la qualité de l'individu qu'il représente. Logiquement, les meilleurs individus ont plus de chances d'être croisés et de participer à l'amélioration de notre population.

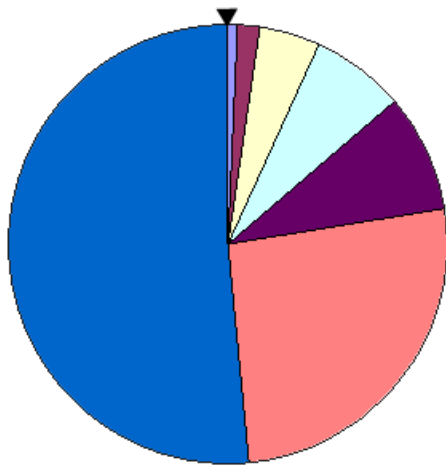


Schéma d'une roulette

A-3. La sélection par tournoi

Le principe de la sélection par tournoi augmente les chances pour les individus de piètre (min) qualité de participer à l'amélioration de la population. Le principe est très rapide à implémenter. Un tournoi consiste en une rencontre entre plusieurs individus pris au hasard dans la population. Le vainqueur du tournoi est l'individu de meilleure qualité. Vous pouvez choisir de ne conserver que le vainqueur comme vous pouvez choisir de conserver les deux meilleurs individus ou les trois meilleurs. À vous de voir, selon que vous souhaitez créer beaucoup de tournois, ou bien créer des tournois avec beaucoup de participants ou bien mettre en avant ceux qui gagnent les tournois haut la main. Vous pouvez faire participer un même individu à plusieurs tournois.

A-4. L'élitisme

Cette méthode de sélection permet de mettre en avant les meilleurs individus de la population. Ce sont donc les individus les plus prometteurs qui vont participer à l'amélioration de notre population. Cette méthode a l'avantage de permettre une convergence (plus) rapide des solutions, mais au détriment de la diversité des individus. On prend en effet le risque d'écarter des individus

de piètre qualité, mais qui aurait pu apporter de quoi créer de très bonnes solutions dans les générations suivantes.

Une autre possibilité relevant aussi du domaine de l'élitisme, pour s'assurer que les meilleurs individus feront effectivement partie de la prochaine génération, est tout simplement de les sauvegarder pour pouvoir les rajouter à coup sûr dans la population suivante.

Les méthodes de sélection permettent de déterminer quels individus nous allons croiser. Reste maintenant à définir comment nous allons les croiser.

B. Les croisements

Les croisements permettent de simuler des **reproductions d'individus dans le but d'en créer des nouveaux**. Il est tout à fait possible de faire des croisements aléatoires. Toutefois, une solution largement utilisée est d'effectuer des croisements multipoints.

-B-1. Les croisements multipoints

Il faut découper en N (2 ou 3 peuvent suffire) morceaux chacun des individus choisis, puis il faut prendre un gène de chaque individu pour créer un nouvel individu.

Notez qu'il est possible de créer ainsi plusieurs individus : si un gène d'un individu ne sert pas à la création d'un individu, il peut servir à la création d'un autre.

Donc en prenant X individus à croiser, vous pouvez potentiellement obtenir X nouveaux individus. Mais rien ne vous empêche d'utiliser plusieurs fois certains gènes, pour obtenir plus de X nouveaux individus.

Notez aussi qu'il n'est pas nécessaire et surtout pas recommandé de croiser tous les individus d'une population, car rien ne nous dit si le résultat d'un croisement sera meilleur ou moins bon que les individus parents.

Individus parents			
Gène1	Gène2	Gène3	Gène4
Gène1	Gène2	Gène3	Gène4
Gène1	Gène2	Gène3	Gène4

Individus enfants			
Gène1	Gène2	Gène3	Gène4
Gène1	Gène2	Gène3	Gène4
Gène1	Gène2	Gène3	Gène4

Problème de l'informaticien en vacances

La méthode de croisement proposée pour ce problème : on commence à faire un croisement « simple » entre deux individus, puis on corrige les individus créés pour qu'ils aient la forme d'une solution.

Par exemple, si nous souhaitons croiser {A,B,C,D,E,F,G,H,I,J} avec {D,A,F,J,C,E,G,H,B,I}, nous pouvons décider que la première moitié du premier parent deviendra la première moitié du

premier enfant, et que la seconde moitié du premier parent deviendra la seconde moitié du deuxième enfant. Et inversement pour le second parent.

Nous obtiendrions ainsi les enfants $\{A,B,C,D,E,E,G,H,B,I\}$ et $\{D,A,F,J,C,F,G,H,I,J\}$. On s'aperçoit tout de suite du problème posé, certaines villes sont visitées 2 fois et d'autres ne le sont pas. Donc, supprimer les doublons et de les remplacer par les villes non visitées.

Ainsi le premier enfant $\{A,B,C,D,E,E,G,H,B,I\}$ serait remplacé par $\{A,B,C,D,E,F,G,H,J,I\}$ ou par $\{A,B,C,D,E,J,G,H,F,I\}$

De même le deuxième enfant passerait de $\{D,A,F,J,C,F,G,H,I,J\}$ à $\{D,A,F,J,C,B,G,H,I,E\}$ ou à $\{D,A,F,J,C,E,G,H,I,B\}$

C. Les mutations

Une autre solution que le croisement pour créer de nouveaux individus est de modifier ceux déjà existants. Une fois de plus, le hasard va nous être d'une grande utilité. Il peut s'avérer efficace de **modifier aléatoirement quelques individus de notre population en en modifiant un gène ou un autre. Rien ne nous dit que l'individu muté sera meilleur ou moins bon, mais il apportera des possibilités supplémentaires qui pourraient bien être utiles pour la création de bonnes solutions.** De même que pour les croisements, **il n'est pas recommandé de faire muter tous les individus.** Il est possible de faire muter un individu de la manière qu'il vous plaira.

Une seule contrainte : **l'individu muté doit être de la forme d'une solution potentielle.**

Généralement, on ne modifie qu'un gène pour passer d'une solution à une autre solution de forme similaire, mais qui peut avoir une évaluation totalement différente.

Problème de l'informaticien en vacances

La méthode de mutation consiste en **une permutation de deux villes**. Nous sommes certains que les individus mutés auront toujours la forme d'une solution potentielle, car nous ne changeons que l'ordre des villes.

Par exemple, $\{A,B,C,D,E,F,G,H,I,J\}$ pourra être muté en $\{A,B,H,D,E,F,G,C,I,J\}$

IV. L'insertion des nouveaux individus dans la population

Une fois que nous avons créé de nouveaux individus que ce soit par croisements ou par mutations, il nous faut sélectionner ceux qui vont continuer à participer à l'amélioration de notre population. Une fois encore, libre au programmeur de choisir ceux qu'il souhaite conserver. Il est possible de refaire une étape d'évaluation des individus nouvellement créés. De même qu'il est possible de conserver tous les nouveaux individus en plus de notre population.

Il n'est toutefois pas recommandé de ne conserver que les nouveaux individus et d'oublier la population de travail. En effet, rien ne nous dit que les nouveaux individus sont meilleurs que les individus de départ.

Une méthode relativement efficace consiste à insérer les nouveaux individus dans la population, à trier cette population selon l'évaluation de ses membres, et à ne conserver que les N meilleurs individus.

A. Comment choisir le nombre N d'individus à conserver ?

Le nombre d'individus N à conserver est à choisir avec soin. En prenant un N trop faible, la prochaine itération de l'algorithme se fera avec une population plus petite et elle deviendra de plus en plus petite au fil des générations, elle pourrait même disparaître. En prenant un N de plus en plus grand, nous prenons le risque de voir exploser le temps de traitement puisque la population de chaque génération sera plus grande.

Une méthode efficace est de toujours garder la même taille de population d'une génération à l'autre, ainsi il est possible de dérouler l'algorithme sur un grand nombre de générations.

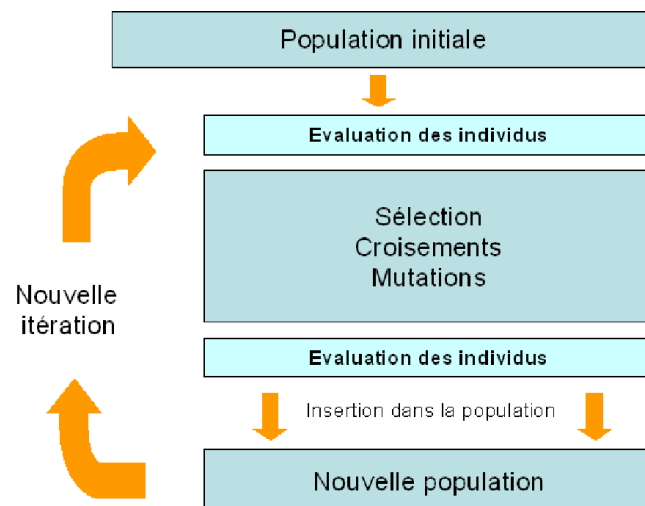
B. Et on passe à la génération suivante

Une fois la nouvelle population obtenue, vous pouvez recommencer le processus d'amélioration des individus pour obtenir une nouvelle population et ainsi de suite ...

V. Réitération du processus

Le programmeur a l'opportunité d'évaluer les individus de sa population avant et/ou après les phases de création d'individus. Le nombre de générations est aussi laissé à l'appréciation du programmeur. Généralement il n'est pas possible de trouver des solutions convenables en moins de 10 générations et au bout de 500 générations, les solutions n'évoluent plus. Mais ceci n'est qu'un ordre de grandeur, tout dépend du problème à résoudre.

Une fois le nombre maximum de générations atteint, vous obtenez une population de solutions. Mais rien ne vous dit que la solution théorique optimale aura été trouvée. Les solutions se rapprochent des bonnes solutions, mais sans plus. Ce n'est pas une méthode exacte.



Conclusion

Un algorithme génétique vous donne une grande liberté dans le paramétrage et dans l'implémentation des différents traitements. Libre à vous ensuite de modifier tel ou tel paramètre si les solutions obtenues ne vous conviennent pas.

Les algorithmes génétiques ont l'énorme avantage de pouvoir être appliqués dans un grand nombre de domaines de recherche de solution, lorsqu'il n'est pas nécessaire d'avoir la solution optimale,

qui prendrait par exemple trop de temps et de ressources pour être calculée (ou tout simplement si personne n'est capable de la trouver de manière théorique).