Les Algorithmes Génétiques

Najet AROUS

Introduction

Les **algorithmes génétiques** (AG) font partie des algorithmes **évolutifs**. Ils sont dus à John Holland et son équipe.

• Les AG sont des **algorithmes d'exploration** fondés sur les mécanismes de la **sélection naturelle** et de la **génétique**. Ils utilisent à la fois les principes de la **survie des structures des mieux adaptées**, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède certaines des **caractéristiques de l'exploration humaine**.

• A chaque génération, un nouvel ensemble de créatures artificielles (des chaînes de caractères) est créé en utilisant des parties des meilleurs éléments de la génération précédente; ainsi que des parties innovatrices.

Introduction

La **théorie d'AG** affirme que dans une population donnée les chromosomes (solutions) les mieux adaptés à l'environnement (évaluation) auront une chance exponentielle élevée de survie et par conséquent une meilleure chance pour produire des progénitures.

De telle méthode de recherche génétique subvient aux besoins de l'exploitation non seulement de meilleures solutions mais aussi de **l'espace de recherche**.

Historique

- Rechenberg (1960) Evolution strategies.
- John Holland (1975), les algorithmes génétiques.

Les **AG** ont été développés pour la première fois par John Holland, à l'université de Michigan. Leurs recherches ont deux objectifs principaux:

1. mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels, et
2. concevoir des systèmes artificiels qui possèdent les propriétés importantes des systèmes naturels. [Gol-91]

Les **AG** constituent une étape importante dans la recherche sur les méthodes d'apprentissage automatique et reçoivent à ce titre l'attention des milieux industriels et des centres d'expérimentation.

- Koza (1992) Programmation génétique.

Historique : L'AG de Holland

1. Les individus sont représentés par une chaîne de bits de longueur fixe *l* exemple, a: individu, a $\in IB^{ll}$ où $IB = \{O,1\}$.
2. La mutation est un événement de renversement de bit qui survient avec une probabilité très faible *Pm* par bit; $Pm \in$[0.005,0.01] Schaffer et al. 1989, $Pm \approx$ 0.01 Grefenstette 1986.
3. L'algorithme utilise un opérateur de recombinaison (crossover) qui échange arbitrairement des sous chaînes entre deux individus avec une probabilité *Pc;* De Jong 1975, *Pc* _∈_[0.75 , 0.95] Schaffer et al. 1989, $Pc \approx$ 0.95 Grefenstette 1986. La longueur et la position des sous chaînes sont choisies aléatoirement, mais sont identiques pour les deux individus.
4. L'opérateur de sélection probabilistique forme la génération suivante en copiant les individus dans la base des probabilités proportion force.

Les Objectifs de l'Optimisation

L'optimisation cherche à améliorer une performance en se rapprochant d'un (ou des) point(s) optimums. Cette définition se compose de deux parties:

❶ nous cherchons une amélioration afin de s'approcher d'un

❷ point optimal.

Il faut donc clairement distinguer le procédé d'amélioration de sa destination, ou l'optimum lui-même.

Cependant, en évaluant les procédures d'optimisation, nous nous concentrons souvent uniquement sur la convergence et oubliant entièrement les performances intermédiaires.

Caractéristiques des AG

Les AG se distinguent des méthodes classiques (énumératives ou basées sur le gradient) de recherche dans un espace d'états car ils :

- utilisent un codage des paramètres du problème et non les paramètres eux mêmes,
- travaillent sur une population et pas sur une unique situation

  ↳ éviter le piège d'un minimum local
- utilisent des valeurs de la fonction (pb) étudiée

  ↳ pas sa dérivée ni une fonction auxiliaire
- utilisent des règles de transition probabilistes

  ↳ pas déterministes

Métaphore

- **But poursuivi** : approcher le maximum d'une fonction.
- **Métaphore de base :**

– Les points du domaine de définition forment une population d'individus.

– La valeur de la fonction en un point **mesure l'adaptation** (***fitness ou score ou force***) de l'individu.

– Les points où le maximum est atteint sont les individus les mieux adaptés (***fittest***).

- **Hypothèse évolutionniste (Darwinisme élémentaire) :**

– Le processus démarre avec une population aléatoire.

– Les individus les mieux adaptés sont produits par évolution de la population (*survival of the fittest*) : les plus fort survivent à la génération suivante, les plus faibles sont éliminés.

– Cette évolution est produite par croisements et mutations.

Terminologie

- Population (= génération):
  - Ensemble d'individus.
- Chromosome (= un individu)
  - Groupe de gènes d'un individu.
- Gène
  - Caractère / caractéristique d'un individu.
- Allèle
  - Forme / valeur prise par une caractéristique.

Principes des AG

Modélisation :

Etats de l'espace de recherche ⇒ chaînes de symboles

individus ou *chromosomes*

Ensemble d'individus ⇒ population

La population évolue au cours de la résolution.

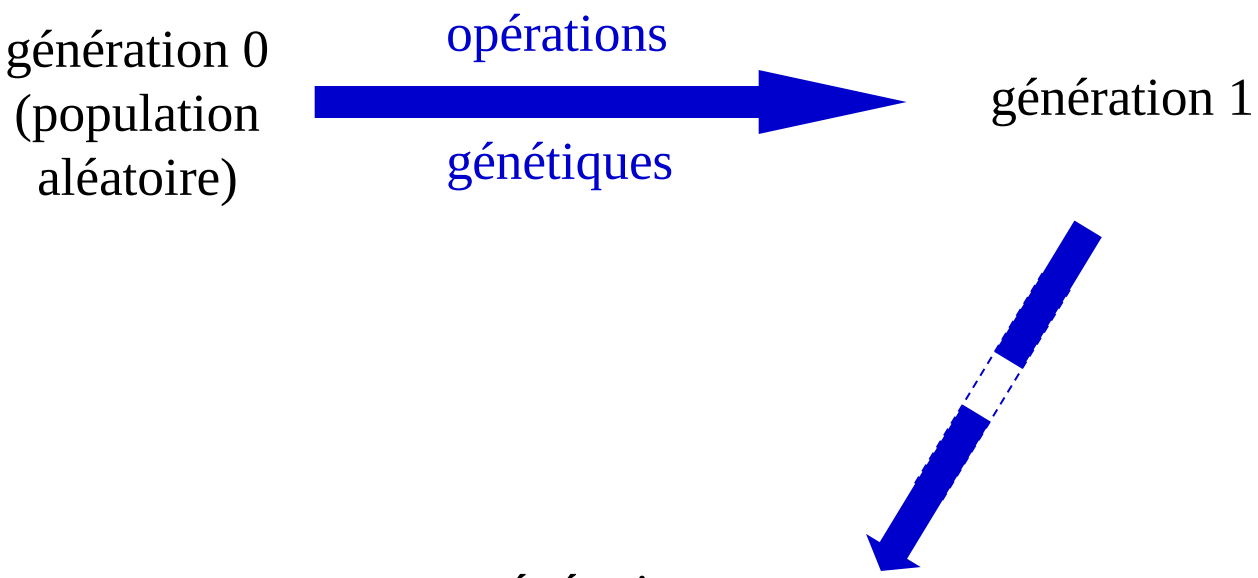
Evolution :

On mesure l'*adaptation* (*fitness*) de chaque individu.

↳ adéquation comme solution

D'une *génération* à l'autre on cherche à conserver les individus les mieux adaptés pour les reproduire (dupliquer) et appliquer à leur descendance des opérateurs dits *génétiques.*

Principes des AG

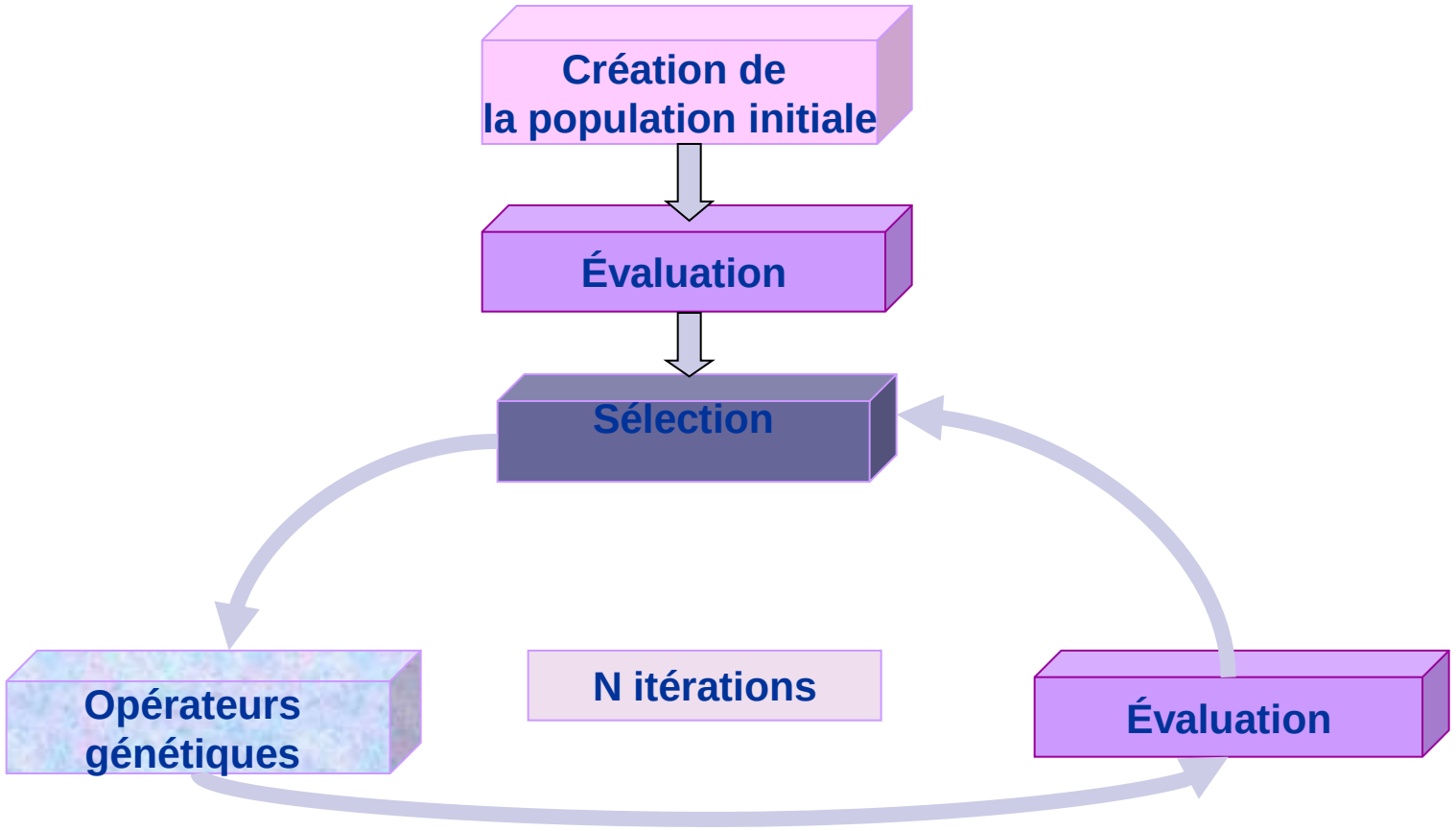


génération *n* :
individus les mieux adaptés =
bonne approximation de la solution du pb

Éléments d’un AG

- Représentation d’une solution par un chromosome.
- Fonction d’aptitude (fitness function).
- Sélection des candidats.
- Opérateurs génétiques.

AG


Création de
la population initiale
Évaluation
Sélection
Opérateurs
génétiques
N itérations
Évaluation

Étapes de Base de l'AG

1. Générer aléatoirement une population de n chromosomes x.
2. Évaluer l'adaptabilité f(x) de chaque chromosome.
3. Créer une nouvelle population en :
    - 3.1 Sélectionnant 2 parents chromosomes.
    - 3.2 Croisant les deux parents pour obtenir un enfant.
    - 3.3 Mutant l'enfant obtenu avec une certaine prob.
    - 3.4 Plaçant l'enfant dans la population.
4. Composer la nouvelle population.
5. Si la nouvelle population n'est pas satisfaisante retourner à 2.

Algorithme de Base : AG Simple

Initialiser la population

Tant que non fini ou non convergence

Calculer le degré d'adaptation de chaque individu

Reproduction des individus

Sélectionner les survivants parmi les parents et les enfants

Appliquer les opérateurs génétiques

Fin tant que

Conclure

Peut être soumis à de nombreuses variantes

AG : Représentation des Chromosomes

Codage binaire

Valeur 0 et 1

Un chromosome est représenté par un vecteur dont les gènes appartiennent à l'ensemble {0,1} (exemple: 1000010011111001101); c'est le codage binaire. De cette façon, nous pouvons décrire une **relation d'appartenance** ou une **matrice d'incidence**. Si la fonction à optimiser possède *n* variables entières, le chromosome sera constitué de la concaténation des *n* codages binaires.

Chromosome A 101100101100101011100101
Chromosome B 111111100000110000011111

☞ **Problème du sac à dos**

- **Exemple : Jouer au tennis**
  - On peut représenter le ciel par trois bits.
  - On peut représenter le vent par deux bits.
  - Le résultat par un seul bit.
  - L'hypothèse « si (vent = faible) jouer = vrai »
    - sera codé : **111 01 1**

AG : Représentation des Chromosomes

➢ Codage par valeur

Type de données quelconque

Chromosome A 1.2324 5.3243 0.4556 2.3293 2.4545

Chromosome B ABDJEIFJDHDIERJFDLDFLFEGT

Chromosome C (back),(back),(right),(forward),(left)

☞Calcul des poids d'un réseau

➢ Codage de permutation

Valeur entière

Chromosome A 1 5 3 2 6 4 7 9 8

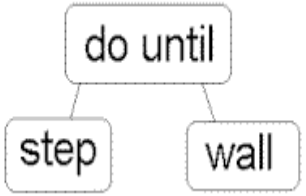
Chromosome B 8 5 6 7 2 3 1 4 9

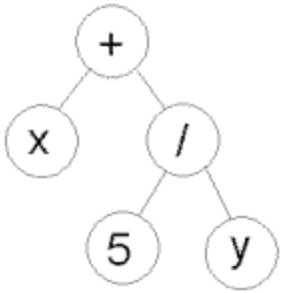
☞ Problème du voyageur de commerce

AG : Représentation des Chromosomes

➢ Codage par listes

Cette technique consiste à représenter une solution par une liste d'éléments (souvent des réels). Elle est fréquemment utilisée pour l'optimisation de certains graphes (recherche du chemin optimal, problème du voyageur de commerce, problème d'ordonnancement des tâches,...). Cette technique est d'une importance capitale pour les problèmes où l'ordre des gènes intervient. Le problème du voyageur de commerce peut être résolu par des chaînes de nombre réels. Chaque noeud représente un identificateur unique. Une solution est alors une concaténation de suite de nombres.





➢ Codage par structure

Structure d'arbre

Construction de la Population Initiale

Tirage au hasard

Les différents chromosomes de la population initiale peuvent être générés de façon aléatoire. Cette méthode permet à la nécessité d'obtenir une population variée facilitant l'exploration des zones diverses de l'espace de recherche.

Heuristiques

La population initiale est initialisée par des solutions provenant des heuristiques, dans l'espoir de les améliorer au cours des générations. Cependant, une telle technique peut influencer l'AG et le faire converger trop rapidement vers un optimum local.

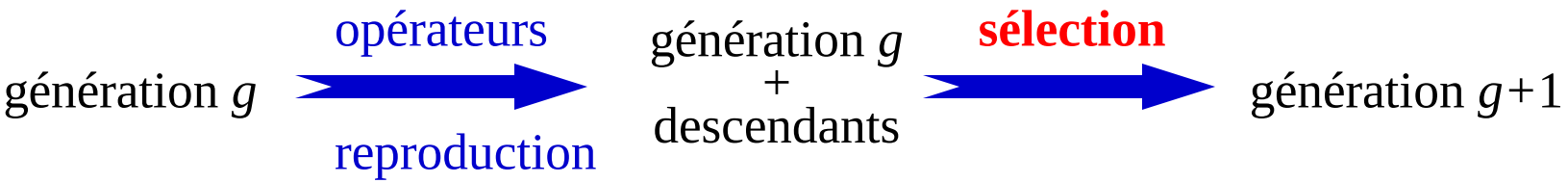
Solutions mixtes: heuristiques + aléatoires

La population initiale est formée de solutions générées par des heuristiques et de solutions aléatoires. Ce qui permet de bénéficier de la notion de variété des solutions.

AG : Fonction d'Adaptation

- La performance de l'AG peut être sensible au choix de la fonction d'adaptation.
- Une fonction d'évaluation permet de :
  - Mesurer à quel point une solution est près de la réponse optimale.
  - Sélectionner les chromosomes pour la prochaine génération:
    - Plus l'aptitude est bonne, plus les chances de survie sont grandes.
    - Il est possible de garder impérativement le chromosome (l'individu) le plus près de la solution optimale.

AG : Sélection



Généralement population de taille constante

Plusieurs stratégies :

- générationnelle : tous les descendants remplacent tous les parents.
- introduction du "***generation gap***" = pourcentage des parents remplacés.

Quels individus conserver ?

Il est naturel de conserver les mieux adaptés :

- utilisation de la roulette biaisée.
- on garde strictement les meilleurs.

Comment choisir les parents?

⇨ Choisir les meilleurs parents en fonction de l'adaptabilité f.

AG : Sélection : Méthode de la Roulette

Pour chaque chromosome *i* on calcule son degré d'adaptation f_i.

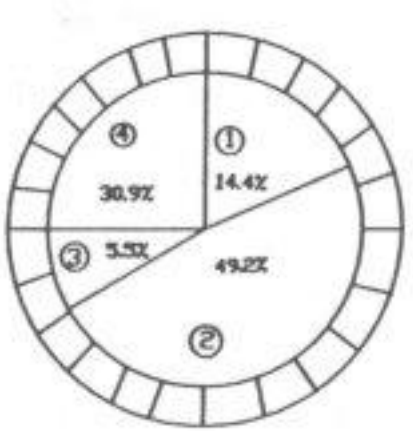
$$p_i = \frac{f_i}{\sum_i f_i} \times 100$$

On crée une roulette biaisée où chaque *i* occupe une portion p_i.

n tirages pour déterminer les descendants d'une génération de taille *n*.

Exemple : optimiser $f(x) = x^2$ et n = 4 individus

i	chaîne	f_i	% total
1	**01101**	169	14.4
2	**11000**	576	49.2
3	**01000**	64	5.5
4	**10011**	361	30.9
Total		1170	100



Lors d'un tirage la chaîne 1 occupe 14.4% de la roue, il y a une probabilité de 0.144 que l'on obtienne une copie de cet individu.

AG : Sélection des Parents

- **Stratégie en état de régime**
  - Sélection de quelques bons parents.
  - Génération des enfants à partir de ces bons parents.
  - Insertion des enfants.
  - Élimination des mauvais chromosomes pour permettre l'insertion des enfants.

- **Stratégie *k-élitiste***

  on garde systématiquement les *k* meilleurs individus d'une génération à l'autre.

AG : Les Opérateurs Génétiques

Les 3 plus courants :

- **Reproduction** : le nombre de descendants d'un chromosome est **proportionnel** à son degré d'adaptation

  ↳ méthode de la **roulette** ou du **tournoi**

- **Croisement** (***cross-over***) : création de deux nouvelles hypothèses (chromosomes) à partir de deux existantes. Croisement 1-point ou 2-point ou uniforme.
- **Mutation** : change un ou plusieurs bits au hasard dans l'hypothèse.

Les taux de croisement et de mutation appliqués lors de la genèse d'une nouvelle population sont des paramètres de l'algorithme à fixer en fonction du problème.

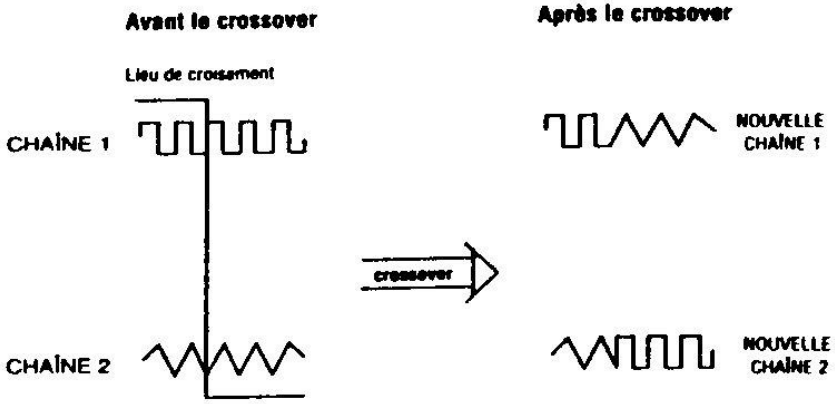
Le taux de mutation est généralement faible.

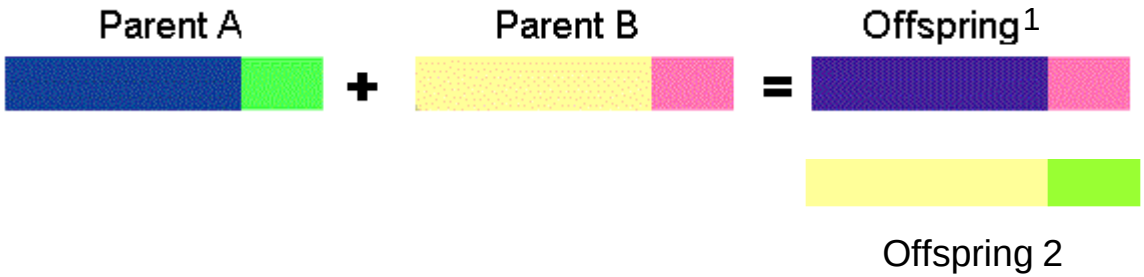
☞ D'autres opérateurs génétiques peuvent être utilisés.

AG : Croisement Simple : 1-point

2 chromosomes de taille *l*.
On choisit aléatoirement un entier *k* entre 1 et *l*-1.
k représente le point de croisement des deux chaînes

On crée 2 nouveaux individus en échangeant les caractères des chaînes initiales compris entre *k+1* et *l*.





AG : Croisement Simple : 1-point

Croisement pour encodage binaire : 1 point de croisement

Sélectionne un emplacement au hasard dans le chromosome et inverse les sections des deux chromosomes.

Exemple 1 : l = 5 et k = 3

$C_1 = 0\ 1\ 1\ |\ 0\ 1$
$C_2 = 1\ 1\ 0\ |\ 0\ 0$

$E_1 = 0\ 1\ 1\ |\ 0\ 0$
$E_2 = 1\ 1\ 0\ |\ 0\ 1$

Exemple 2 : l = 16 et k = 5

Chromosome 1 :11011 | 00100110110

Chromosome 2 :**11011** | **11000011110**

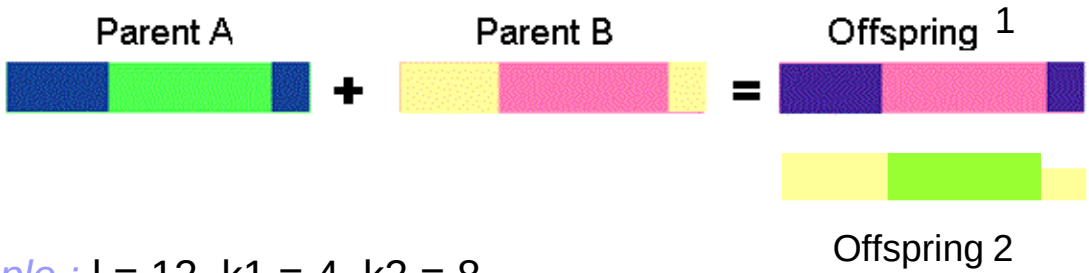
Enfant 1 : *11011* | **11000011110**

Enfant 2 : **11011** | *00100110110*

AG : Croisement Double : 2-points

- Même principe que le croisement simple mais sélectionne deux points et applique le croisement sur la section entre les deux points.

2 points de croisement



Exemple : l = 12, k1 = 4, k2 = 8

P_1: 110011001100

P_2: 011010110101

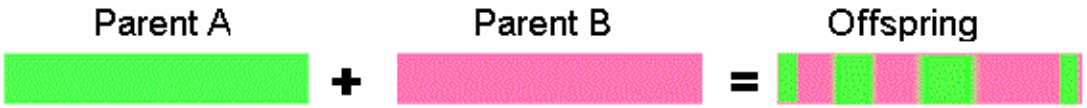
Produit

E_1:110010111100

E_2:011011000101

AG : Croisement Uniforme

- Sélectionne des bits au hasard et applique le croisement sur ces bits.



Exemple : l = 12, k1 = 2, k2 = 5, k3 = 6, k4 = 9

P_1: 110011001100

P_2: 011010110101

Produit

E_1: 111011110100

E_2: 010010001101

AG : Mutation

Il s'agit de la modification aléatoire de la valeur d'un caractère de la chaîne.

Pour un codage binaire, elle consiste simplement à changer un 0 en un 1 et réciproquement.

Le taux de mutation est généralement choisi très faible (≈ 1 pour 1000)

↳ pour chaque caractère des descendants, probabilité de 1/1000 qu'il mute

AG : Mutation

Sélectionne un ou plusieurs bits au hasard et inverse leur valeur.

Exemple 1 : encodage binaire

Original enfant 1 :1101111000011110
Original enfant 2 :1101100100110110

Mutant enfant 1 : 1100111000011110
Mutant enfant 2 : 1101101100110110

P_1: 110011001100 —Produit→ E_1: 110001001100

Dans ce cas, le bit 5 à été inversé.

Exemple 2 : encodage réel

2 nombres sont choisis et échangés

(1 **2** 3 4 5 6 8 9 7) => (1 **8** 3 4 5 6 2 9 7)

Paramètres des AG

Croisement

Espérance d'amélioration des nouvelles générations.
Probabilité varie de 0 à 100%.
0 % (pas de croisement) clones parfaits.
100% pas de clones.

Mutation

Sortir de minima locaux.
Inverser un zéro avec un un.
Probabilité variant de 0% à 100%.
Probabilité normalement faible.

Taille de la population

Petite taille limite l'exploration de l'espace.
Grande taille réduit la vitesse de convergence.

Problèmes de Conception

Le codage

• La première question qui se pose est le codage des points du domaine sous forme de chaînes de symboles.

• Ce codage doit être *compatible avec les mécanismes évolutifs* (croisement et mutation) qui seront définis en conséquence.

• Exemple : problème du voyageur de commerce : étant donné les distances entre les *n* villes, trouver un parcours reliant les *n* villes qui soit le plus court possible.

Problèmes de Conception

Méthode de sélection

• Comment réaliser la sélection en favorisant les individus les plus aptes, mais sans tomber dans un élitisme stérilisant ?

• La sélection "comme à la roulette" (*roulette wheel selection*) attribue à chaque chromosome une probabilité de sélection proportionnelle à son aptitude (*fitness*) :

– si certains chromosomes sont beaucoup plus aptes que les autres, ces derniers n'ont aucune chance...

• La sélection "suivant le rang" attribue à chaque chromosome une probabilité de sélection proportionnelle à son *rang* par ordre d'aptitude :

– même les moins aptes ont une chance...

Problèmes de Réglages

La mise au point des paramètres d'un AG est une tâche très difficile et nécessite un temps intensif. Pendant plusieurs années les paramètres appliqués dans le domaine des AG étaient ceux proposés par De Jong.

Cependant, il s'est avéré que les paramètres optimaux dépendent d'un problème à un autre. Ces paramètres dépendent aussi de la taille de la population, de la représentation binaire, et des opérateurs génétiques classiques.

Les paramètres optimaux varient d'un problème à un autre, mais les paramètres robustes s'exécutent bien à travers une variété de problèmes.

- Longueur des chaînes ?
- Taille de la population ?
- Nombre de générations ?
- Choix des probabilités de mutation (faible) et de croisement ?

Problèmes de Réglages

Il existe une technique adoptive pour déterminer ses paramètres rapidement et efficacement. Elle peut être utilisée pour déterminer les meilleures valeurs des paramètres pour les algorithmes non traditionnels de haute performance destinés aux domaines spécifiques. Cependant, la majorité des praticiens des AG préfèrent l'AG traditionnel [Dav91].

Taille de la population *(N):* la taille de la population influe sur la performance globale et l'efficacité de l'AG. Les AG avec des petites populations ne s'exécutent pas comme d'habitude de façon efficace parce que la population fournit une étendue insuffisante de l'espace du problème. Une population nombreuse est plus vraisemblable à être représentative du domaine entier du problème. De plus, une population nombreuse empêche la convergence prématurée aux solutions globales au lieu de locales; cependant, elle augmente sensiblement le temps d'exécution.

Problèmes de Réglages

L'écart de génération (G): *N*G* structures d'une population *P(t)* sont choisies pour être remplacées dans *P(t+1).* Une valeur de G=1 signifie que la population entière est remplacée à chaque génération.

Taux de croisement *(Pc):* contrôle la fréquence d'application de l'opérateur de croisement. Chaque nouvelle population, *N*Pc* chromosomes subissent un croisement plus le taux de croisement est élevé, plus les nouvelles structures sont introduites rapidement dans la population.

Problèmes de Réglages

Taux de mutation *(Pm):* permet d'augmenter la variabilité de la population. Un taux faible de mutation empêche une position donnée de geler à une seule valeur; un taux de mutation élevé résulte essentiellement dans une recherche aléatoire. Approximativement *Pm*N*L* mutations surviennent chaque génération, où *L* est la longueur d'un chromosome.

Stratégie de sélection (S): plusieurs stratégies de sélection sont possibles. Une stratégie de sélection pure où chaque structure dans la population actuelle est reproduite selon son adaptation. Une stratégie d'élitisme: la sélection est exécutée en premier lieu et les structures avec les meilleures performances sont choisis pour survivre à la prochaine génération.

Exemple pas à pas

Optimiser $f(x)=x^2$ pour x entre 0 et 31 population de n= 4 individus

Choix du codage du paramètre : x codé en binaire sur l=5 caractères.

un individu $\in \{0,1\}^5$

Degré d'adaptation : on peut utiliser f directement dans ce cas.

Génération initiale et pré-calculs

n°	chaîne	x	$f(x)$	p_i
1	0 1 1 0 1	13	169	0,14
2	1 1 0 0 0	24	576	0,49
3	0 1 0 0 0	8	64	0,06
4	1 0 0 1 1	19	361	0,31
Total			1170	1
moyenne			293	
max			576	

Croisement

individu	partenaire	position croisement	obtenu	x	$f(x)$
0 1 1 0 | 1	2	4	0 1 1 0 | 0	12	144
1 1 0 0 | 0	1	4	1 1 0 0 | 1	25	625
1 1 | 0 0 0	4	2	1 1 | 0 1 1	27	729
1 0 | 0 1 1	3	2	1 0 | 0 0 0	16	256
	Total				1754
	Moyenne				439
	Max				729

On essaie une mutation de 0,001 et rien n'est modifié.

Il faut maintenant sélectionner les survivants et recommencer.

Algorithmes génétiques (Exemple 1)

- Exemple: Étant donnés les chiffres [0..9] et les opérations +,-,*, /, utiliser un algorithme génétique pour trouver la séquence qui aura comme résultat une valeur cible.
- Supposons que la valeur cible est 20
  - Une solution possible est 4+9*4/2-2
- Représentation des hypothèses
  - On peut utiliser 4 bits pour former toutes les possibilités
- Fonction d'aptitude:
  - Nous voulons une valeur élevée

  Lorsqu'une solution est près de la cible. Donc:

$$Aptitude(x) = \frac{1}{x - cible}$$

0	0000	8	1000
1	0001	9	1001
2	0010	+	1010
3	0011	-	1011
4	0100	*	1100
5	0101	/	1101
6	0110		
7	0111		

Problème du Voyageur de Commerce

- Problème du voyageur de commerce (PVC) :
  - Un voyageur doit passer par toutes les villes d'une région.
  - Il ne peut passer deux fois par la même ville.
  - Termine son voyage à la ville de départ.
- On cherche à minimiser la distance parcourue par le voyageur.

Le problème

Trouver le trajet de longueur minimale passant par toutes les villes et revenant au point de départ (distance euclidienne)

Problème du Voyageur de Commerce

- Recherche exhaustive
  - Il suffit de construire tous les chemins possibles et de calculer leurs longueurs. Avec n villes il y a (n - 1)!/2 chemins possibles. Avec 36 villes on trouve:516657398319307246483332566876160000000
  - Cette méthode est donc inutilisable, sauf si le nombre de villes est très petit.
- Les méthodes
  - Méthodes exactes « Branch and bound » [*record 13509 villes*].
  - Méthodes approchées: Algorithme gloutons,
  - Méthodes de descente,
  - Recuit simulé,
  - Algorithme tabou,
  - Algorithmes génétiques,
  - Algorithme de Lin et Kerningham,
  - Colonies de fourmis, [*quelques millions de villes*].

PVC : Algorithmes Génétiques

1- On code la solution du problème à résoudre sous la forme de gènes (*fonction d'encodage*)

2- On génère une population d'individus aléatoirement (*initialisation*)

3- On teste les individus et on les fait mourir si leur génome n'est pas bon (*fitness/sélection*)

4- On croise les survivants et on retourne en 3.

On recommence tant que le génome des survivants n'est pas une solution satisfaisante au problème

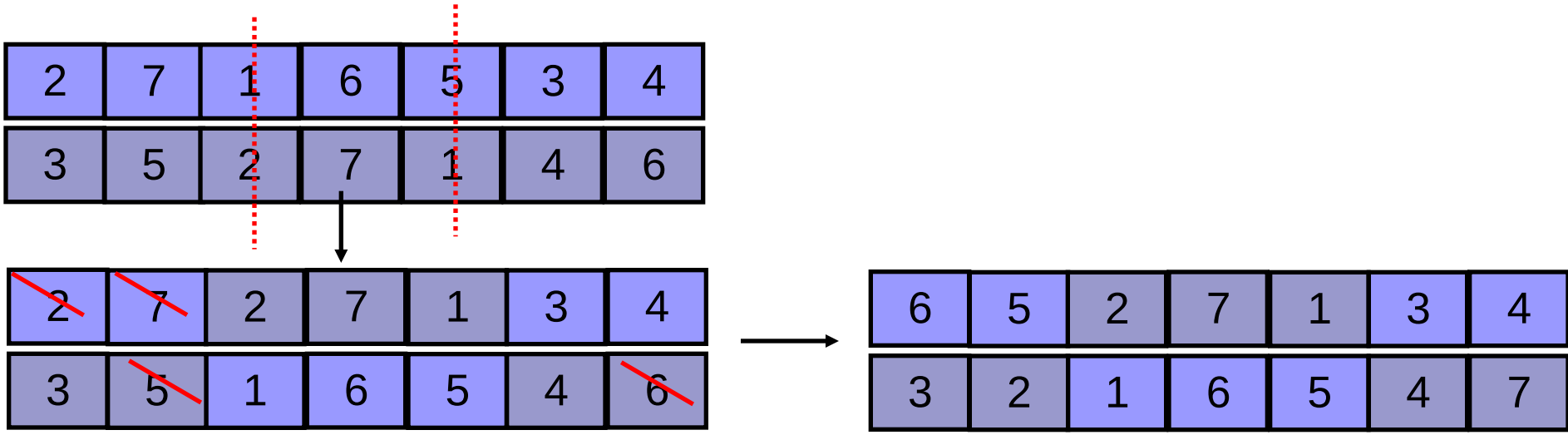
Représentation d'une hypothèse (chromosome)

- Dans un vecteur de dimension n (pour n villes) chaque case représente une ville, l'ordre de parcours est de la case 1 à n.

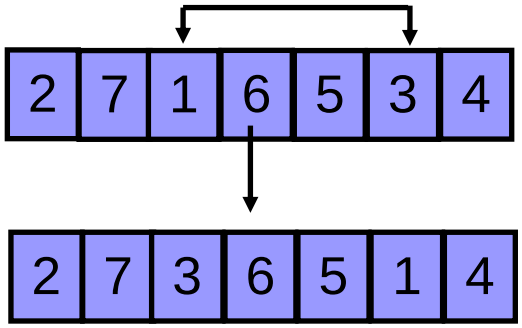
2	7	1	6	5	3	4

PVC : Algorithmes génétiques

- Croisement: croisement par deux points



- Mutation : échange deux villes dans le vecteur



Conclusion

- Les **AG** de part leur robustesse représentent une méthodologie viable pour **les problèmes d'optimisation et d'exploration**.
- Les **AG** ont fait la preuve de leur capacité dans de nombreuses études théoriques et expérimentales.
- Les **AG** produisent des échanges de notions novatrices entre les individus, et sont donc liés à l'idée que nous nous faisons du **processus humain de recherche et de découverte**.
- Les **AG** manipulent les représentations des variables de décision ou de contrôle au niveau de l'individu pour tirer parti des similarités entre les individus performants.

Conclusion

- Les **AG** réalisent une optimisation efficace en ne tenant compte que des valeurs de la fonction. Ils traitent simultanément les similarités dans le codage sous-jacent et l'information qui permet d'ordonner les structures en fonction de leur capacité de survie dans l'environnement considéré. En exploitant un type d'information si généralement disponibles, les **AG** peuvent être appliqués virtuellement à tous les problèmes.
- Les **AG** utilisent le hasard pour guider une exploration qui tire grandement parti de l'information disponible.
- Ne trouve pas toujours la solution optimale.
- Utilisable lorsque l’espace de recherche est très grand.
- Peuvent être utilisés comme algorithme d’apprentissage.

Références

John Holland (U. of Michigan, 1975) *Adaptation in Natural and Artificial Systems.*
David E. Goldberg (élève de Holland) *Genetic Algorithms* 1991 traduit en français par Vincent Corruble, *Algorithmes génétiques,* Paris 1994.

Beaucoup de références sur Internet :
http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html
mais aussi :
Modern Heuristic Techniques for Combinatorial Problems, Edited by C.R. Reeves, Backwell Scientific Publications.