

## COURS SUR

# L'APPRENTISSAGE ARTIFICIEL

## COURS MASTER IFI 2010/2011



JEAN-DANIEL ZUCKER

DR À L'IRD UR GEODES  
(MODÉLISATION MATHÉMATIQUES ET INFORMATIQUES DES SYSTÈMES COMPLEXES)  
UMMISCO UMI 209



## *Administratif: 1/2 Module Apprentissage (18ECTS)<sup>2</sup>*

- **Séance 1: Jeudi 25 Novembre** – INTRO GÉNÉRALE
  - Introduction, principe inductif, historique, formulation
  - Quelques mots sur l'apprentissage statistique
  - Espace des versions et algorithme
- **Séance 2: Lundi 6 Décembre** – APPRENTISSAGE SUPERVISÉ
- **Séance 3: Mardi 7 Décembre** – APPRENTISSAGE NON-SUPERVISÉ
- **Séance 4: Mardi 11 Janvier 2011** – ALGORITHMES ÉVOLUTIONNAIRES
- **Séance 5: Jeudi 14 Janvier 2011** – ALGORITHMES PAR RENFORCEMENT
- **Séance 6: Lundi 17 Janvier 2011** – MINI-PROJET

# Algorithmes Évolutionnistes

1. Introduction
2. Résultats théoriques
3. Applications
4. Programmation Génétique
5. Conclusion

# Algorithmes Évolutionnistes

## 1- Introduction

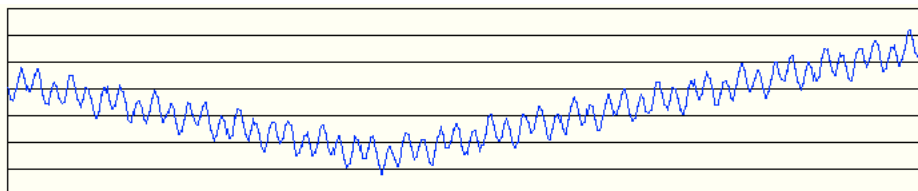
### Plan

- Inspiration
- Terminologie
- La fonction de performance
- L'algorithme canonique
- Opérateurs de reproduction
- Exemple de fonctionnement

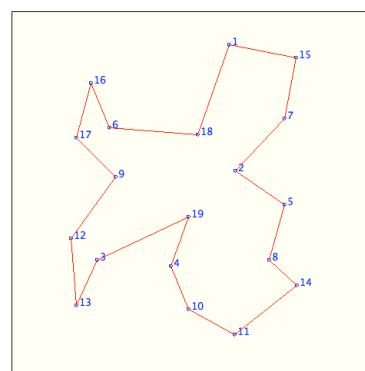
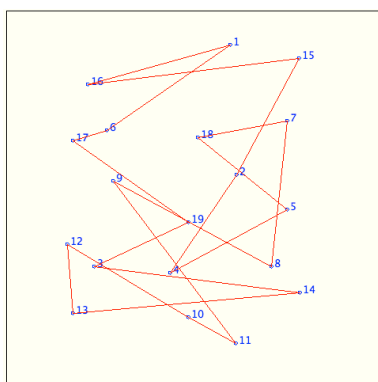
## Introduction : attrait de la métaphore de l'évolution

- Recherche dans de gigantesques espaces

- Intérêt du parallélisme, théorie Darwinienne.
- Intérêt de méthodes d'optimisation globales
- Quels Pbs ? Espaces « non informés » « Aiguille dans 1 botte de foin »
- Quand ? Une fois que l'on a essayé toutes les autres méthodes
- Recherche de comportement bio-inspirés (Animat, Biomorph, ...)



## Introduction : motivation problèmes NP-difficiles



(ex: le voyageur de commerce)

<http://www.obitko.com/tutorials/genetic-algorithms/tsp-example.php>

## I-Terminologie (1/3)

7

- **Chromosome** : chaîne d'ADN, ou de bits, etc. codant le **génotype**. Un chromosome est constitué de plusieurs gènes : unités fonctionnelles codant des caractères particuliers.

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

- **Individu** : Un élément  $I$  de l'espace  $\Omega$  des possibilités ayant un chromosome.
- **Génotype** : Ensemble de gènes codant l'individu. (son « programme »)
- **Phénotype** : Ensemble des caractères ou traits de l'individu. Expression du génotype. (son « execution »)
- **Population  $\Pi(t)$**  : Collection des individus en compétition (éventuellement en coopération aussi) à l'instant  $t$ . Collection de  $P$  éléments de  $\Omega$ .

COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## I-Terminologie (2/3)

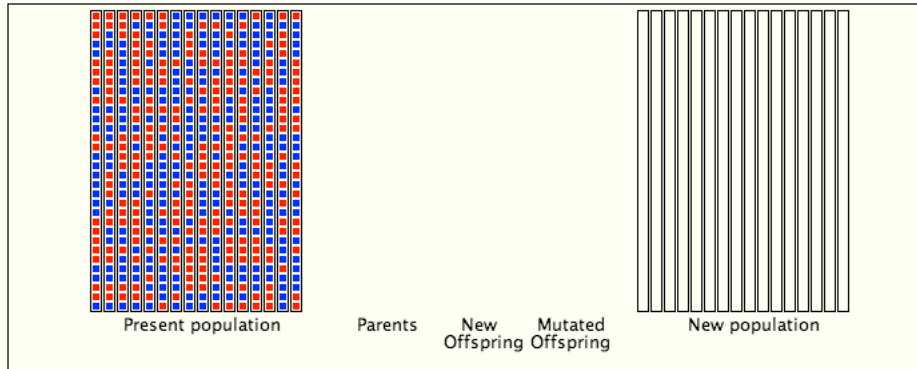
8

- **Performance ("fitness")** : Valeur de l'individu  $I$  suivant la fonction  $F$  dont on cherche un optimum (fonction de performance définie sur  $\Omega$ ). La qualité de l'individu  $I$  mesuré d'après son phénotype (la « qualité de son programme »). Ex:
  - Génotype = programme pour voler
  - Phénotype = exécution pendant  $T$  minutes
  - Fitness = durée du temps de vol
- Cette performance force les individus à être en compétition pour le privilège de se reproduire.

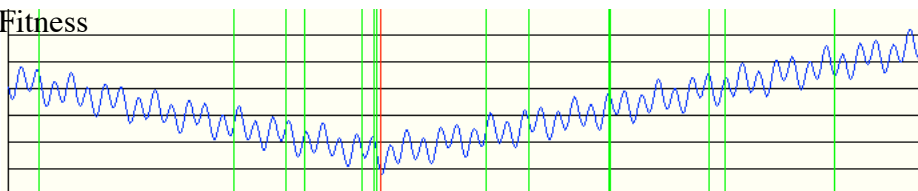
COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## Population



## Fitness

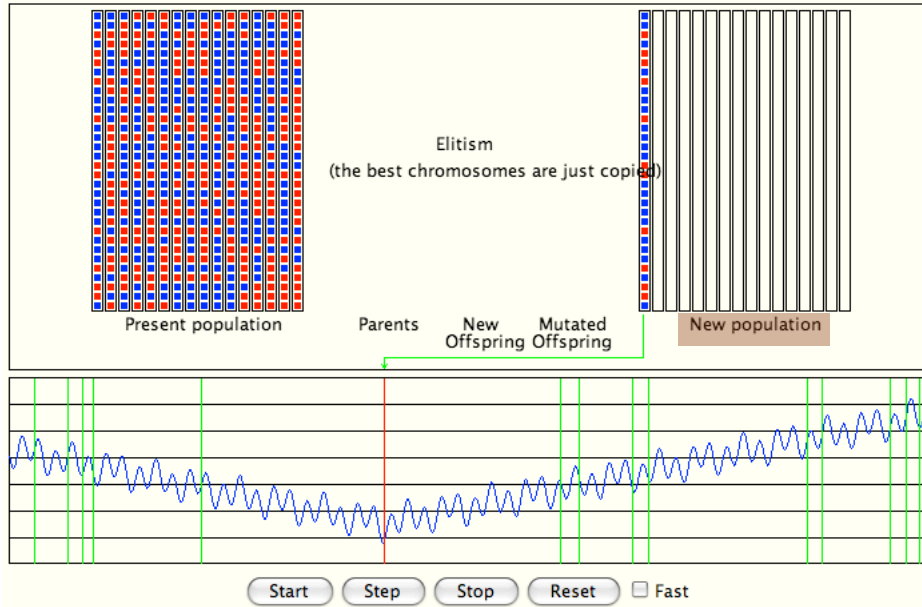


## I-Terminologie (3/3)

- **Génération** : Passage de la population  $\Pi(t)$  à  $\Pi(t+1)$ .
- **Reproduction** : Calcul par des opérateurs de nouveaux individus à partir d'individus de  $\Pi(t)$ . Ce calcul s'effectue sur le génotype.
- **Croisement** : Un opérateur de reproduction :  $\Omega \times \Omega \rightarrow \Omega$ .
  - Inspiration : la reproduction sexuée. Essayer de profiter des meilleurs gènes des deux parents.
  - Produit un chromosome en échangeant des gènes des deux parents. (les « offsprings »)
- **Mutation** : Un opérateur de reproduction :  $\Omega \rightarrow \Omega$ .
  - Inspiration : la reproduction asexuée.
  - Permet de diversifier le matériel génétique au-delà de celui de  $\Pi(t)$ .

## Reproduction: *élitisme*

11

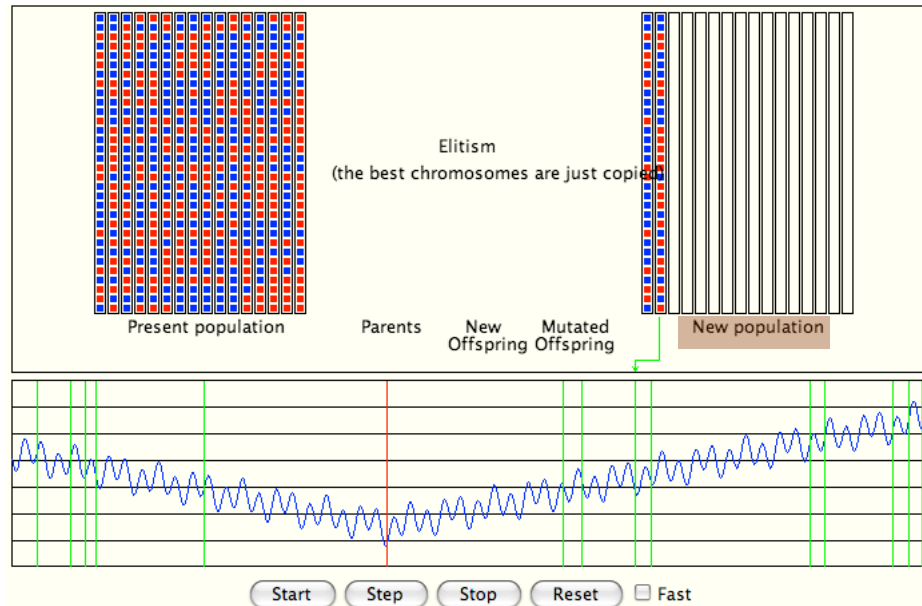


COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## Reproduction: *élitisme*

12



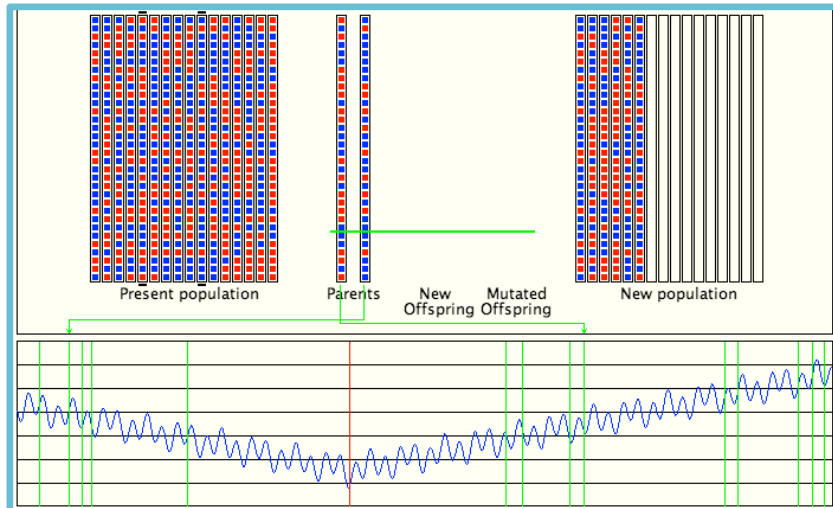
COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## Reproduction: croisement

13

Chromosome 1	11011   00100110110
Chromosome 2	11011   11000011110
Offspring 1	11011   11000011110
Offspring 2	11011   00100110110



COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## Reproduction: croisement

14



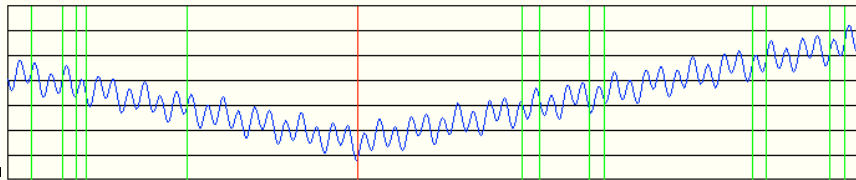
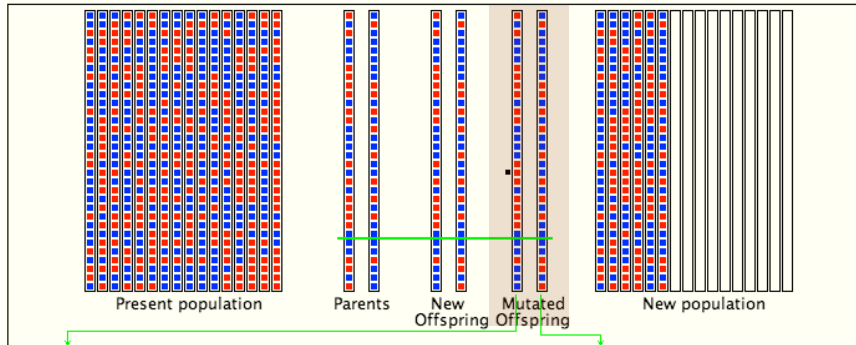
COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

# Mutation

15

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

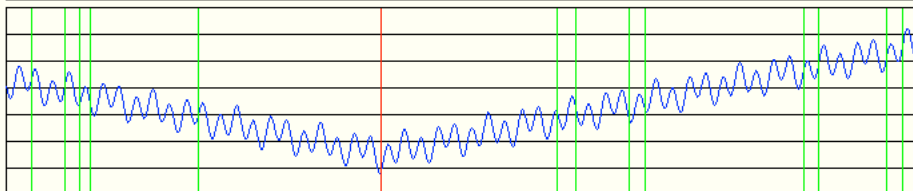
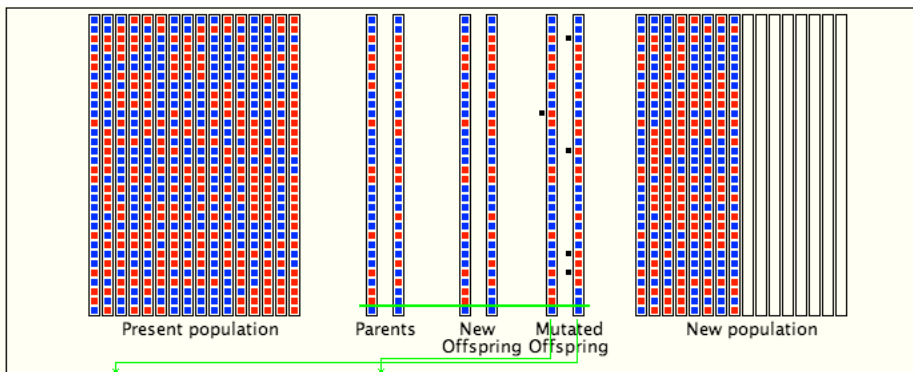


COURS API

IFI 2011

# Mutation

16

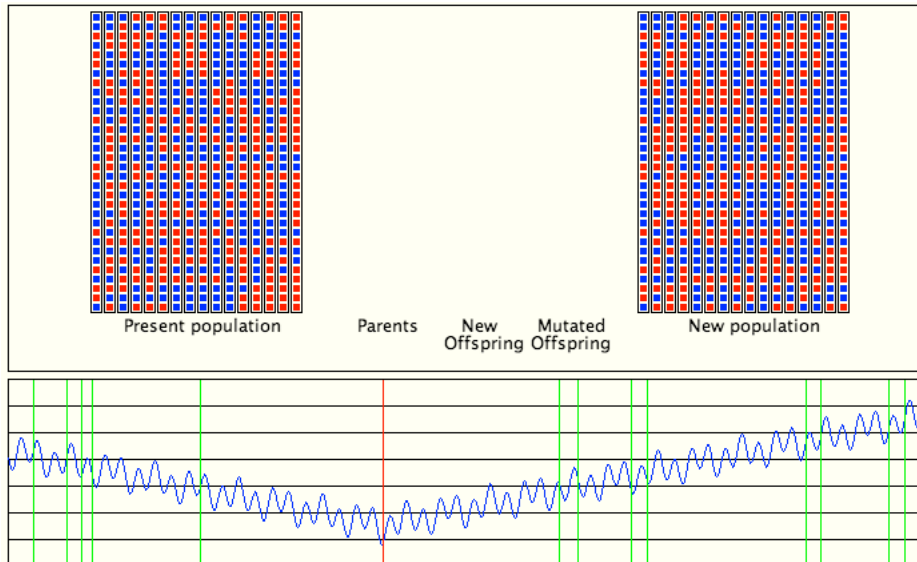


COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## Une nouvelle population

17

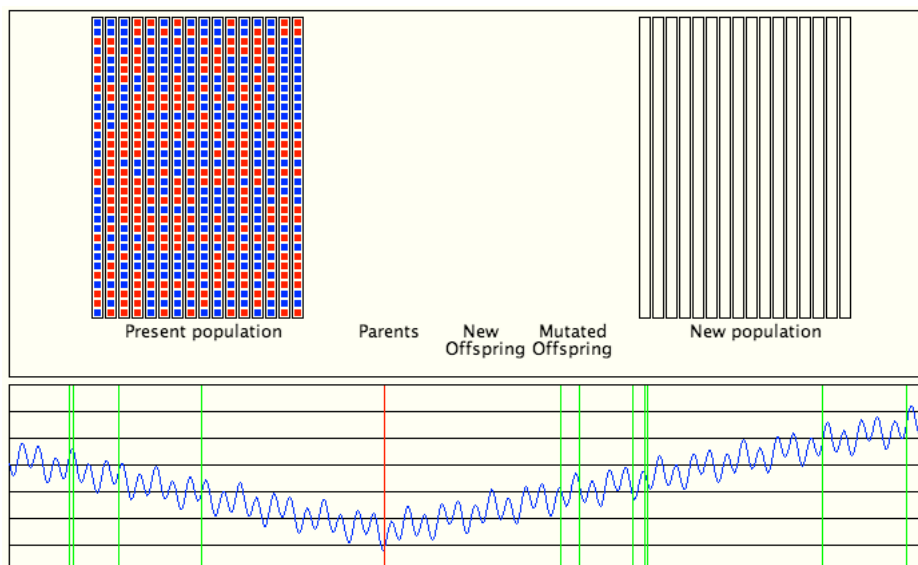


COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## On recommence...

18

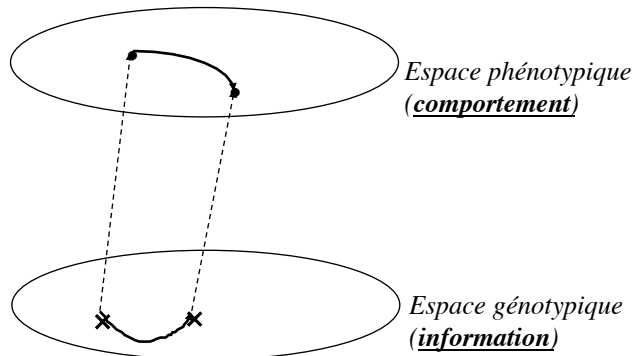


COURS APPRENTISSAGE N°4 Jean-Daniel ZUCKER

IFI 2011

## Relations phénotypes - génotypes

- Espace **phénotypique** : Evaluation / Sélection
- Espace **génotypique** : Opérateurs génétiques
  - croisement, mutation)



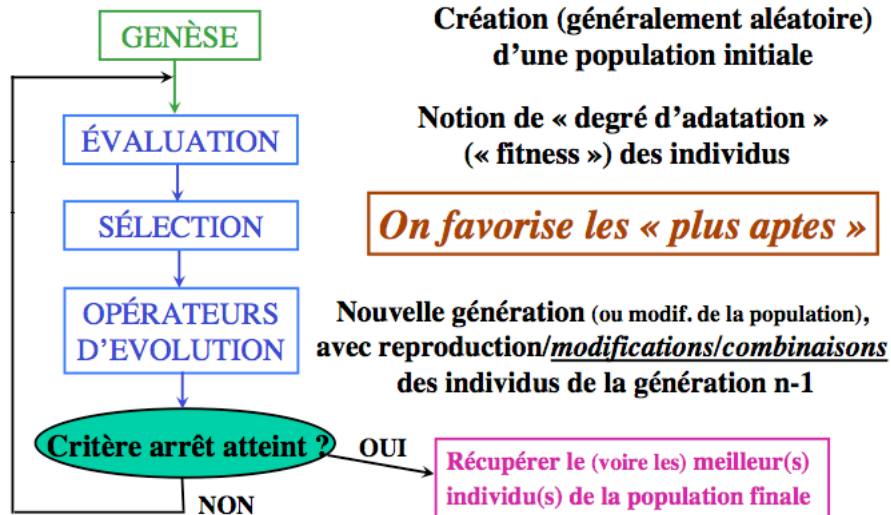
### Question :

Comment avoir une bonne représentation (espace génotypique) et de bons opérateurs pour obtenir une exploration judicieuse de l'espace phénotypique ?

## I- L'algorithme canonique (1/2)

- **Idée générale** : Utiliser le principe de l'évolution darwinienne des populations pour optimiser une fonction de performance.
- **"Les plus aptes (performants) survivent"**.
- **Processus** :
  - Les individus mutent et se reproduisent
  - Les plus aptes sont sélectionnés et forment la génération suivante amenée à se reproduire.
  - Au bout d'un certain nombre de générations, des individus particulièrement aptes (performants) apparaissent dans la population.

## 1- Algorithme évolutionniste canonique



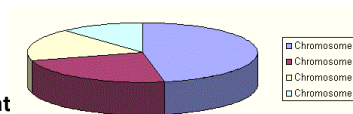
## 1- L'algorithme génétique canonique

Cas particulier de l'algo. évolutionniste où chaque individu est représenté par un «génom»(séquence de bits, ou d'entiers, de réels,...)

1. **Initialisation** de la population  $\Pi(0)$
2. **Évaluation** de la population  $\Pi(t)$  courante.  
Calcul de  $F(X)$  pour tout  $X$  de  $\Pi(t)$
3. **Sélection** des meilleurs individus (les **parents**) selon la fonction de fitness  $F$ .
4. **Reproduction** des parents par **croisements** et **mutations**
5.  $\Pi(t+1) = \text{Remplacement}(\Pi(t), \text{Enfants})$   
(On peut se limiter aux  $N$  meilleurs et  $\Pi(t+1) = N \text{ Enfants} + [\Pi(t) - N \text{ pires}]$ )
6. Si le **critère d'arrêt** non satisfait, retour en 2.

### (Etape 3) Sélection (exploitation)

- **Objet:** Choisir ceux qui se reproduisent
- **Objectif: Favoriser les meilleurs**
  - Excès -> perte de diversité
  - Insuffisance -> pas de convergence
- **Modes:**
  - **A) Déterministe:** (p parents donne e enfants)
    - on prend les p meilleurs parmi e (résultats conv.)
    - on prend les p meilleurs parmi p+e (robustesse prat)
  - **B) Stochastique,** par tirage à la roulette
    - 1) Selon la fraction de fitness  
(pb: les mauvais ont des chances)
    - 2) Selon le rang
  - **C) Stochastique,** par tournoi



### B.1) Roulette sur la fraction de fitness



- **Sélection proportionnelle à la performance**
  - **Roulette de Monte-Carlo**  $P_{Sélection} = \frac{F(\text{individu})}{\sum_{\Pi(t)} F(\text{individu})}$ 
    1. Calcul de la somme des performances des individus : S
    2. Répéter N fois (N individus)
      - Tirer un entier r entre 0 et S
      - Faire la somme partielle des individus jusqu'à celui qui la fait passer > r.  
Le sélectionner.
  - **Variante de l'échantillonnage universel stochastique [Baker, 87]**
  - **Inconvénient : favorise trop et trop tôt les bons individus**
    - Convergence prématurée
    - ➔ Méthode de renormalisation

## B.2) Roulette sur le rang



25

- **Sélection par le rang**
  - **Principe :**
    - Les individus sont ordonnés suivant leur performance
    - Leur espérance de sélection est proportionnelle à leur rang
  - **Avantage : Plus “doux” que sélection proportionnelle**  
→ Convergence moins rapide
  - **Inconvénient : On perd l’information sur la valeur absolue des individus**
    - Moindre pression sélective quand la variance des performances est grande
    - Plus forte pression quand la variance est faible

## C) Stochastique par tournoi

26

### Sélection par tournoi

- **Idée : éviter les calculs sur toute la population (les sommes ou les tris)**
- **Principe :**
  - **Tirer au hasard deux individus**
    - Tirer aléatoirement un nombre  $r$  entre 0 et 1
      - Si  $r < k$  (paramètres de l’algo) alors sélectionner le plus performant des 2 individus
      - Sinon, sélectionner le moins performant.
  - **Recommencer autant de fois que nécessaire (tirage avec remise)**

## (Etape 4a) Croisement

<http://www.obitko.com/tutorials/genetic-algorithms/tsp-example.php>

Analogie: reproduction sexuée

Intuition; les enfants héritent des qualités de leurs deux parents...

Croisement **binaire**: échange de bits entre parents

- a) Croisement à 1 point (Holland)
 

1 0 0 0 1	1 0 1 0 0 1 1 0 1 1	→ (a)	1 0 0 0 1 0 0 1 0 1 1 1 1 0 1
0 0 1 0 1	0 0 1 0 1 1 1 1 0 1		0 0 1 0 1 1 0 1 0 0 1 1 1 0 1
- b) Croisement à 2 points (Dejong)
 

1 0 0 0	1 1 0 1 0 0 1 1	0 1 1	→ (b)	1 0 0 0 1 0 0 1 0 1 1 1 0 1 1
0 0 1 0	1 0 0 1 0 1 1 1	1 0 1		0 0 1 0 1 1 0 1 0 0 1 1 1 0 1
- c) Croisement uniforme (Syswerda)
 

1 0 0 0 1 1 0 1 0 0 1 1 0 1 1	→ (c)	1 0 1 0 1 0 0 1 0 1 1 1 1 1 1
0 0 1 0 1 0 0 1 0 1 1 1 1 0 1		0 0 0 0 1 1 0 1 0 0 1 1 1 0 0 1

Croisement **réel**: croisement barycentrique:  $aX + (1-a)Y$

Croisement **destructeur**: attention à l'influence du codage !

## (Etape 4b) La mutation

Analogie: reproduction asexuée

Intuition: les enfants du croisement sont limités par la population courante, seul contrepoids: la mutation.

Grandes lignes:

- *L'enfant doit être en général proche du parent*
- *L'enfant doit pouvoir être n'importe où*

Pour chaque parent et chaque position: proba de mutation  $p_m$

$(b_1, b_2, \dots, b_N) \xrightarrow{p_m} (b_1, b_2, \dots, b_l, \dots, b_N)$

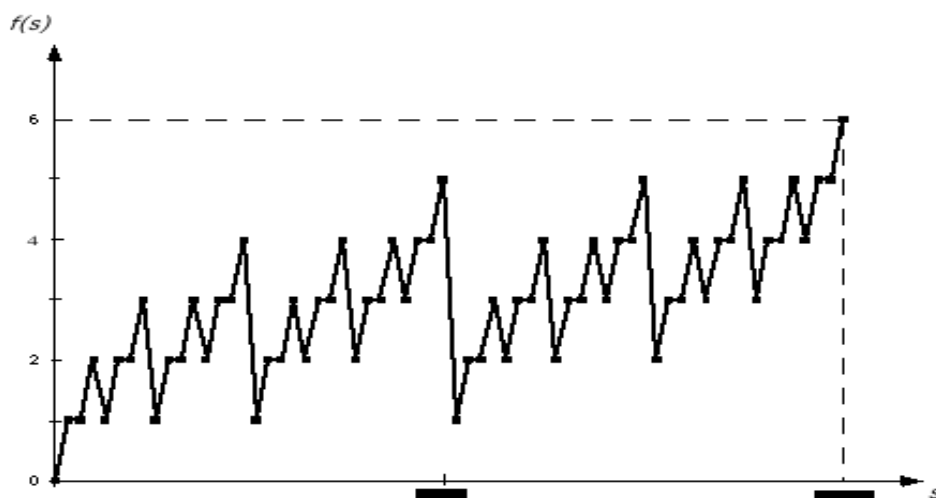
Pour une mutation effective:  $p_m > 1/N \times P$

## Le débat croisement — mutation

- Faire de grands pas:
  - plus rapide plus risqué
- Croisement = mutation contrôlée par la diversité de la population
- En fin d'évolution:
  - le croisement est presque inactif
  - la mutation est de plus en plus destructrice

## 1- Exemple (1/6)

- Optimisation de la fonction "One-Max" (nb de bits à 1)



## 1- Exemple (2/6)

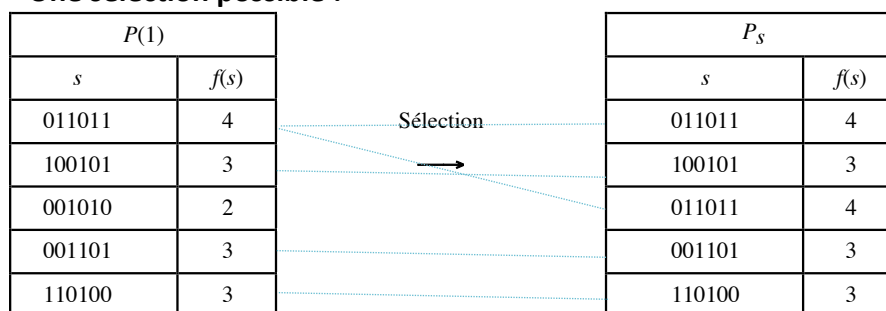
- Taille population = 5. Fitness moyenne = 3

$\Pi(1)$		
$s$	$f(s)$	$P_{s\acute{e}lect}(s)$
011011	4	$\frac{4}{4 + 3 + 2 + 3 + 3} \approx 27\%$
100101	3	20%
001010	2	13%
001101	3	20%
110100	3	20%

Individus; fitness; Probabilité de sélection (par la méthode de sélection proportionnelle à la fitness)

## 1- Exemple (3/6)

- Une sélection possible :



## 1- Exemple (4/6)

- Le croisement ( $p_c = 0.8$ )

Individus sélectionnés pour le croisement	
s	f(s)
011011	4
100101	3
011011	4
110100	3

Opérations :

$$\left. \begin{array}{l} 0110|11 \\ 1001|01 \end{array} \right\} \rightarrow \left\{ \begin{array}{l} 0110|01 \\ 1001|11 \end{array} \right.$$

$$\left. \begin{array}{l} 01|1011 \\ 11|0100 \end{array} \right\} \rightarrow \left\{ \begin{array}{l} 01|0100 \\ 11|1011 \end{array} \right.$$

## 1- Exemple (5/6)

- Après l'opération de croisement

$\Pi_S$		Croisement →	$\Pi_r$	
s	f(s)		s	f(s)
011011	4		011001	?
100101	3		100111	?
011011	4		010100	?
001101	3		001101	3
110100	3		111011	?

## 1- Exemple (6/6)

- La mutation : ( $p_m = 0.05$ )

$\Pi_r$	
$s$	$f(s)$
011001	?
100111	?
010100	?
001101	3
111011	?

Mutation  
→

$\Pi(2)$	
$s$	$f(s)$
<u>1</u> 11001	4
1001 <u>0</u> 1	3
01010 <u>1</u>	3
001101	3
111011	5

Nouvelle performance moyenne =  $18/5 \approx 3.6$

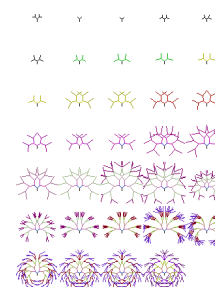
## 2. Quelques résultats théoriques

R. Cerf (1994) : avec «Théorie des schémas»+ processus stochastiques (chaînes de Markov), a montré dans cas des AG à génomes binaires :

1. Quelle que soit la population initiale, l'AG converge (si la taille de population est «assez grande»)
2. Si on ne conserve pas systématiquement à l'identique le meilleur individu d'une génération dans la suivante, alors la limite n'est PAS l'optimum !!
3. Si au contraire on impose cet «élitisme», alors l'AG converge bien vers l'optimum GLOBAL [mais on ne sait pas en combien de temps...]
4. La convergence est assurée par les mutations, même sans croisement (le rôle de ceux-ci seraient donc surtout d'accélérer la convergence)

### 3. Applications

- Systèmes dynamique non-linéaires
- Prédiction
- Analyse de données
- Construire des réseaux de neurones,
- Architecture et contrôleur de robots et animats.
- Planification et Stratégie
- Découverte de la forme de molécules
- Ordonnancement et séquençement (emploi du temps, voyageur de commerce, etc.)
- Fonctions pour la création d'image (ex: biomorph); interface utilisateurs (visualisation)

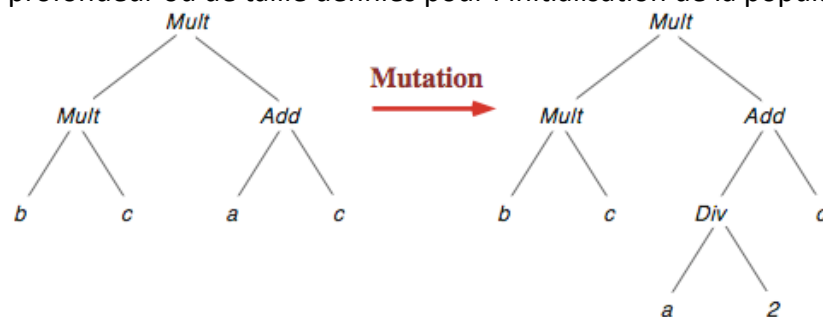


### 4- Programmation Génétique [Koza,92,95]

- $F: \Omega \rightarrow \mathbb{R}$  ;  $\Omega = \text{Arbres}(\mathcal{N}, \mathcal{T})$  (Noeuds, Terminaux)  
(pgms Lisp)
- Exemples :
  - Expressions mathématiques définies sur  $\mathbb{R}$ 
    - $\mathcal{N} = \{+, *, -, \text{exp}, \text{log}, \dots\}$
    - $\mathcal{T} = \{X, Y, \dots, N\}$
  - Programmes
    - $\mathcal{N} = \{\text{if-then-else}, \text{while-do}, \dots\}$
    - $\mathcal{T} = \{\text{Expressions}, \text{Affectations}\}$
  - Modèles rhéologiques
    - $\mathcal{N} = \{+, //\}$
    - $\mathcal{T} = \{\text{Ressort}(k), \text{Amort}(\eta), \text{Patin}(\sigma_s)\}$

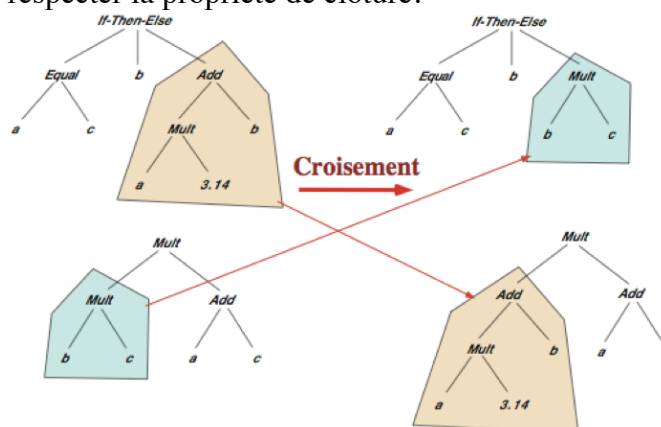
## Programmation Génétique

**La mutation.** C'est un opérateur défini de  $\Omega \rightarrow \Omega$  opérant donc sur un seul génome et produisant un autre génome par altération aléatoire du premier. Plus précisément, la mutation sur une représentation par arbre opère en sélectionnant au hasard un sous-arbre et en le remplaçant par un autre sous-arbre engendré aléatoirement, en respectant les limites de profondeur ou de taille définies pour l'initialisation de la population.



## Programmation Génétique

Le croisement est un opérateur défini de  $\Omega \rightarrow \Omega \rightarrow \Omega \rightarrow \Omega$  produisant deux génomes à partir de la donnée de deux génomes en entrée. Il sélectionne au hasard un sous-arbre dans chacun des génomes fournis en entrée et les échange pour produire deux nouveaux génomes. Il est évident que cette opération doit respecter la propriété de clôture.



## Programmation Génétique

- Protocole expérimental

- $N = \{+, *, -, /, \exp, \log\}$
- $T = \{X, Y, \dots, N\}$
- Population : 500 à 2000
- # exemples : 200 à 2000
- # générations : 50
- fitness = taux de classification ou  $\Sigma$  des distances à la séparatrice
- Sélection proportionnelle à la fitness.  $P_c = 90\%$ ,  $P_\mu = 0\%$

- Résultats :

- Très haut taux de succès
  - jusqu'à environ espace à 20 dimensions
  - et séparatrices jusqu'à 10 variables (pertinentes)
- Rq : Utiliser un outil de simplification des équations trouvées

## Programmation Génétique

- En pratique

- Problème du choix de  $\mathcal{N}$  et de  $\mathcal{T}$
- Problème de l'arité des noeuds, du typage des noeuds
- Problème de l'initialisation de la population
  - Profondeur limite des arbres
  - Tirage aléatoire dans  $\mathcal{N}$  et dans  $\mathcal{T}$
- Problème de la profondeur limite des arbres
- Problème de récursivité, définition de sous-fonctions, ...

- Analyse

- Grande expressivité ( $\Omega$  très riche)
- Pourquoi / comment ça marche ?

*(Beaucoup d'expériences sur des problèmes jouets, et peu d'analyse)*



## **Conclusion (en bref)**

- **Quand utiliser l'évolution artificielle ?**
  - ❑ **Espace de recherche très important**
  - ❑ **Fonction de performance accidentée (non dérivable, non continue) ou bruitée**
  - ❑ **Optimum multiples**
  - ❑ **Pas de nécessité de trouver un optimum global**

## **Annexe: Mise en œuvre des AG et GP (exemples)**

En JAVA:

Package JAVA: A Java-based Evolutionary Computation Research System

<http://cs.gmu.edu/~eclab/projects/ecj/>

En Netlogo

<http://www.cs.northwestern.edu/~fjs750/netlogo/final/gpdemo.html>

En R

Packages: gafit, mcga, genalg

En GAMA

## ***Annexe: applications de GP dans l'industrie***

- \* Solve IT Software  
[http://www.solveitsoftware.com/?content=technology/genetic\\_programming.jsp](http://www.solveitsoftware.com/?content=technology/genetic_programming.jsp)
- \* Natural Selection, Inc.  
[http://www.natural-selection.com/tech\\_1.html](http://www.natural-selection.com/tech_1.html)
- \* NuTech Solutions  
[http://www.nutechsolutions.com/tech\\_gp.asp](http://www.nutechsolutions.com/tech_gp.asp)
- \* RML Technologies, Inc.  
<http://www.rmltech.com/technology%20overview.htm>
- \* Quantum Trader  
<http://www.quantumtrader.com/AI.aspx>
- \* Gepsoft Ltd.  
<http://www.gepsoft.com/gxpt.htm>
- \* Evolved Analytics  
[http://www.evolved-analytics.com/capabilities\\_services.html](http://www.evolved-analytics.com/capabilities_services.html)