

Département d'Informatique Fondamentale et ses Applications
DIFA

Dr .Esma BENDIAB
Maître de conférences
Master 2 SDIA



CHAPITRE 1 :
LES SYSTÈMES ÉVOLUTIFS ET
ADAPTATIFS COMPLEXES

Les systèmes évolutifs et adaptatifs complexes (SEAC)



- Représentent une classe de systèmes dynamiques qui évoluent en fonction de leurs interactions internes et de leur environnement.
- Ces systèmes se caractérisent par une multitude d'entités interagissant de manière non linéaire, souvent en l'absence de contrôle centralisé, aboutissant à des comportements émergents, imprévisibles et auto-organisés.
- On retrouve des exemples de SEAC dans plusieurs disciplines, comme l'écologie, la biologie, l'économie, la sociologie, et même dans l'informatique.

Exemples de Systèmes Évolutifs et Adaptatifs Complexes



Écosystèmes :

- Les écosystèmes sont un exemple classique de SEAC.
- Les interactions entre les organismes (prédateurs, proies, plantes, etc.) et leur environnement évoluent constamment.
- Chaque organisme s'adapte à son environnement, influençant ainsi les autres espèces et les ressources disponibles.

Marchés financiers :

- Les marchés sont composés de nombreux acteurs (investisseurs, banques, entreprises) qui prennent des décisions basées sur une information imparfaite.
- Le marché évolue de manière imprévisible en fonction de ces interactions, ce qui mène à des fluctuations constantes et parfois des comportements de crise (comme des krachs boursiers).

Exemples de Systèmes Évolutifs et Adaptatifs Complexes



Internet et réseaux sociaux :

- Les réseaux informatiques et sociaux sont des systèmes adaptatifs où les individus ou les nœuds se connectent et interagissent, créant des dynamiques globales qui ne sont pas prévisibles en regardant uniquement les interactions individuelles.

Évolution biologique :

- L'évolution darwinienne est un autre exemple de système complexe.
- Les individus et les espèces changent par le processus de mutation, sélection naturelle et dérive génétique, conduisant à des changements à long terme dans la composition de populations.

Les algorithmes évolutionnaires : Préambule

5

- Les Algorithmes Evolutionnaires (AEs) font partie du champ de l'intelligence artificielle.
- Ils sont apparus à la fin des années 1950.
- Il s'agit d'une Intelligence de bas niveau inspirée de l'intelligence naturelle.
- basés sur la théorie de l'évolution et de la sélection naturelle élaborée par Charles Darwin,

Théorie de l'évolution: 'Dans un environnement quelconque, seules les espèces les mieux adaptées survivent, les autres étant condamnées à disparaître'.

- Au sein de chaque espèce, le renouvellement des populations est essentiellement dû aux meilleurs individus de l'espèce

Les algorithmes évolutionnaires : Préambule

6

- Les AEs constituent une approche originale :
 - il ne s'agit pas de trouver une solution analytique exacte, ou une bonne approximation numérique,
 - mais de trouver des solutions satisfaisant au mieux certains critères, souvent contradictoires.
- S'ils ne permettent pas de trouver à coup sûr la solution optimale
- Mais, les solutions fournies sont généralement meilleures que celles obtenues par des méthodes plus classiques, pour un même temps de calcul,

Les algorithmes évolutionnaires : Préambule

7

- Trois classes d'AEs ont été développées indépendamment, entre la moitié des années 1960 et 1970:
 - les algorithmes génétiques
 - la programmation évolutionnaire
 - les stratégies évolutionnaires

Algorithmes génétiques



- Les **algorithmes génétiques (AG)** sont une méthode heuristique inspirée de la théorie de l'évolution de Darwin. appartenant à la classe des algorithmes évolutionnaires.
- Ils sont utilisés pour résoudre des problèmes d'optimisation en s'appuyant sur des mécanismes biologiques tels que la sélection naturelle, la mutation, et le croisement.
- Les AG sont particulièrement efficaces pour explorer de grands espaces de solutions, souvent en présence de nombreuses variables et de contraintes complexes.

Fondements Biologiques

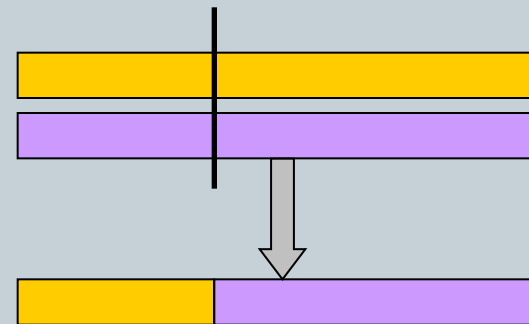
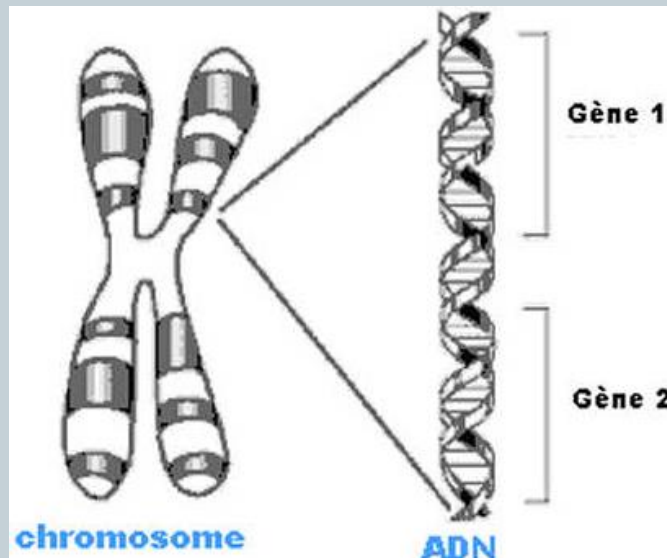


- Les AG sont inspirés par :
- **Sélection naturelle** : Les individus les plus adaptés survivent et se reproduisent.
- **Reproduction** : Deux individus échangent leurs gènes pour produire une nouvelle génération.
- **Mutation** : De légers changements aléatoires dans le génome permettent d'introduire de la diversité.
- **Fitness** : Une mesure de la qualité ou de l'adaptation d'un individu à son environnement.

Fondements Biologiques

10

- Algorithmes génétiques
 - Gène et génotype
 - Crossing over



Algorithmes génétiques: l'inspiration

11

- Algorithmes génétiques

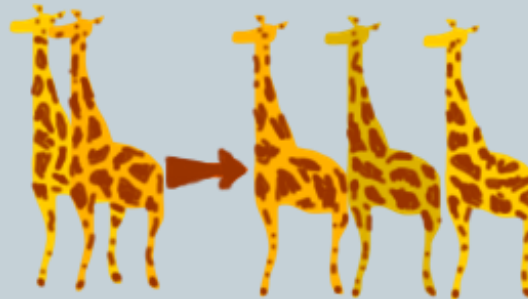
- Individus différents



- Sélection des mieux adaptés



- Hérité



Définitions

12

- D1. Un "algorithme génétique" est défini par un individu/chromosome/séquence et une solution potentielle au problème donné.
- D2. Une "population" est un ensemble de chromosomes ou de points de l'espace de recherche
- D3. "L'environnement" est assimilé à l'espace de recherche
- D4. La fonction que nous cherchons à optimiser est appelée "fonction de fitness, qui associe un coût à chaque solution.

Propriétés

13

- Pour résumer, Lerman et Ngouent (1995) distinguent quatre principales propriétés qui font la différence fondamentale entre ces algorithmes des autres méthodes :
- P1. Les algorithmes génétiques utilisent un codage des paramètres, et non les paramètres eux-mêmes.
- P2. Les algorithmes génétiques travaillent sur une population de points, au lieu d'un point unique.
- P3. Les algorithmes génétiques n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
- P4. Les algorithmes utilisent des règles de transition probabilistes, et non déterministes.
- La simplicité de leurs mécanismes, la facilité de leur mise en application et leur efficacité même pour des problèmes complexes ont conduit à un nombre croissants de travaux ces dernières années.

Principe de fonctionnement des algorithmes génétiques

14

- Un AG opère sur une population d'individus représentant des solutions potentielles du problème à résoudre.
- A chaque génération, l'algorithme applique les trois opérations suivantes en 4 phases
 1. La sélection : choisir les individus les plus performants pour la reproduction et la survie.
 2. La reproduction : permet le brassage, la recombinaison et les variations des caractères héréditaires des parents pour former les descendants.
 3. L'évaluation : offre à chaque individu une qualité d'adaptation qu'on appelle 'fitness'.

La sélection est appliquée en premier lieu pour choisir des individus à reproduire, et puis elle est appliquée à la fin de l'itération pour choisir les individus qui vont survivre,

Principe de fonctionnement des algorithmes génétiques

15

- Ces algorithmes sont itératifs dont le but est d'optimiser une fonction « fitness », Ils sont fondés sur les principes suivants:
- Une représentation chromosomique des solutions du problème.
- Une méthode pour générer une population initiale de solutions.
- Une fonction d'évaluation qui joue le rôle de l'environnement, elle classe les solutions en fonction de leurs aptitudes.
- La reproduction qui consiste en des opérateurs génétiques qui définissent la manière dont les caractéristiques des parents sont transmises aux descendants

Représentation des Solutions



- **Chromosomes**

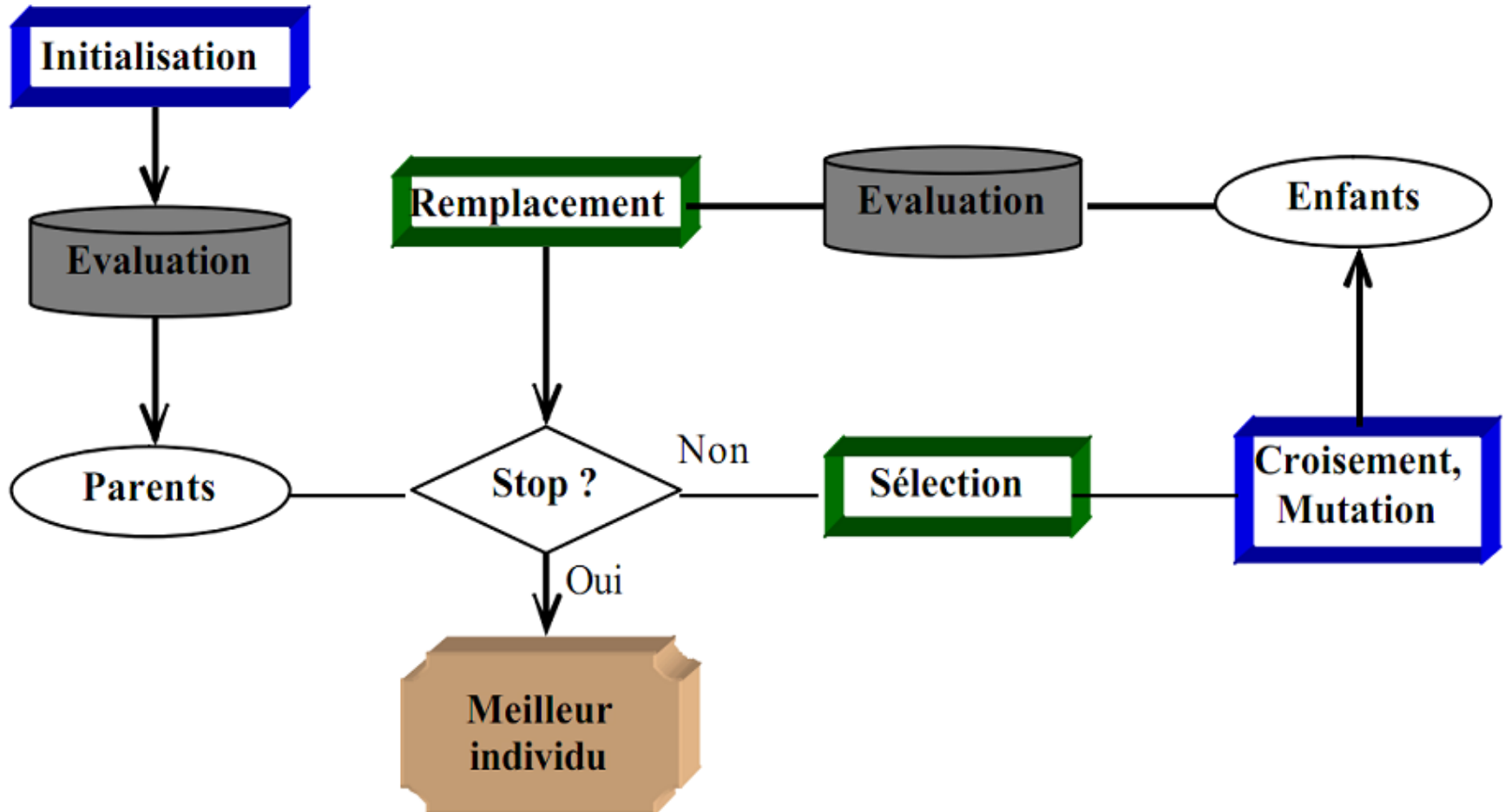
- Les solutions à un problème sont représentées par des **chromosomes**, qui sont souvent des chaînes binaires ou des vecteurs de nombres réels. Chaque chromosome correspond à une solution potentielle dans l'espace de recherche.

- **Génération initiale**

- Une population initiale de chromosomes est générée aléatoirement. Ces chromosomes représentent les points de départ pour l'évolution des solutions.

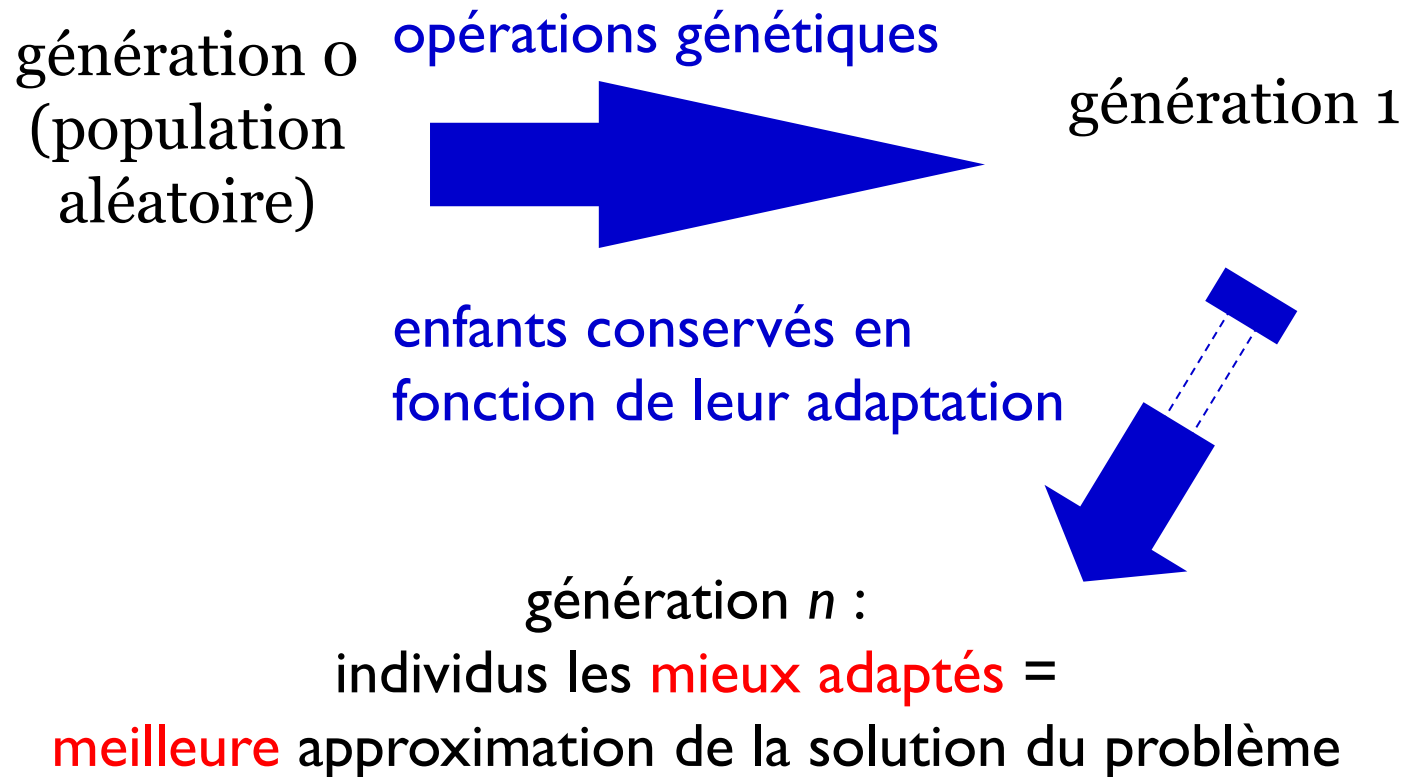
Principe de fonctionnement des algorithmes génétiques

17



Algorithme de base

18



Algorithme

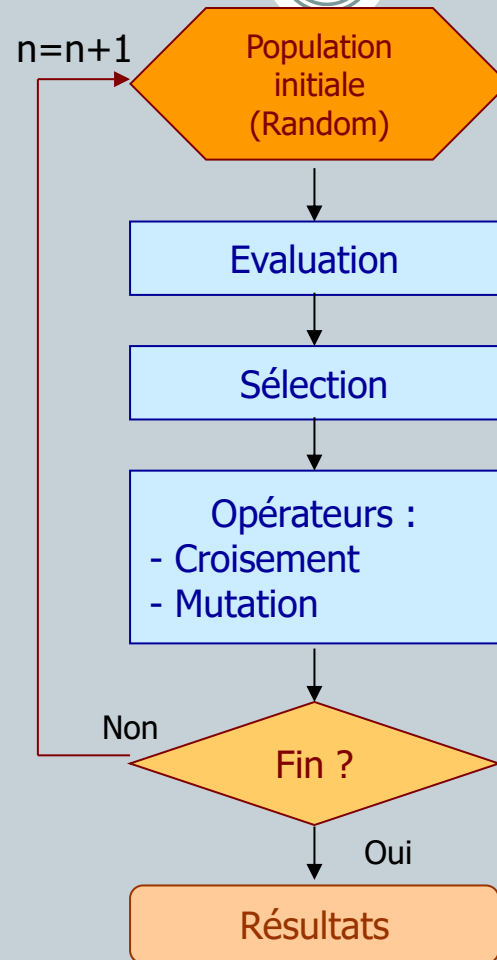
19

initialiser la population (générer aléatoirement une population de N chromosomes x)
calculer le degré d'adaptation $f(x)$ de chaque individu
Tant que non fini ou non convergence
 reproduction des parents
 sélectionner 2 individus à la fois
 appliquer les opérateurs génétiques
 calculer le degré d'adaptation $f(x)$ de chaque enfant
 sélectionner les survivants parmi les parents et les enfants
fin Tant que
conclure

peut être soumis à de nombreuses variantes

Organigramme

20



Fonctionnement d'un Algorithme Génétique



- Les AG suivent généralement un cycle composé des étapes suivantes :

Les opérateurs de bases des algorithmes génétiques

22

- Les algorithmes génétiques s'appuient sur trois opérations de base:
 - La sélection : permet de favoriser les individus qui ont une meilleure fitness (souvent, c'est la valeur de la fonction objectif de la solution associée à l'individu).
 - Le croisement : combiner deux solutions parents pour former un ou deux enfants (offspring) en essayant de conserver les bonnes caractéristiques des solutions parents.
 - La mutation : Elle permet d'ajouter de la diversité à la population en mutant certaines caractéristiques (gènes) d'une solution

Les opérateurs génétiques

23

Les 3 plus courants :

- **reproduction** : le nombre de descendants d'un chromosome suit son degré d'adaptation
 - ↳ méthode de la **roulette**, du **rang** ou du **tournoi**
 - **croisement** (« **crossover** ») : 1-point ou 2-point
 - **mutation**
-
- Les taux de croisement et de mutation appliqués lors de la genèse d'une nouvelle population sont des paramètres de l'algorithme
 - Le taux de mutation est généralement faible
 - On travaille généralement à population constante.

- Les différentes méthodes de sélection

La Sélection



- Des individus sont sélectionnés pour se reproduire en fonction de leur **fitness**.
- Les individus ayant une meilleure fitness ont une plus grande probabilité d'être choisis.
- Les méthodes de sélection incluent :
 - **Roulette** : Sélection proportionnelle à la fitness.
 - Rang ;
 - **Tournoi** : Un sous-ensemble de la population est tiré au sort, et le meilleur est sélectionné.

Sélection: Méthode de la roulette

26

Pour chaque chromosome i on calcule son degré d'adaptation f_i et on pose :

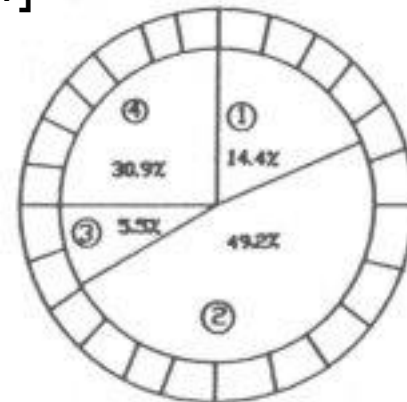
$$p_i = \frac{f_i}{\sum_i f_i} \times 100$$

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

Pour déterminer les descendants d'une génération de taille n , il faut n tirages

exemple : maximiser $f(x) = x^2$ avec $n = 4$ pris dans $[0, 31]$

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 de loterie, il y a une probabilité de 0.144 que l'on obtienne une copie de cet individu

Sélection: Méthode du rang

27

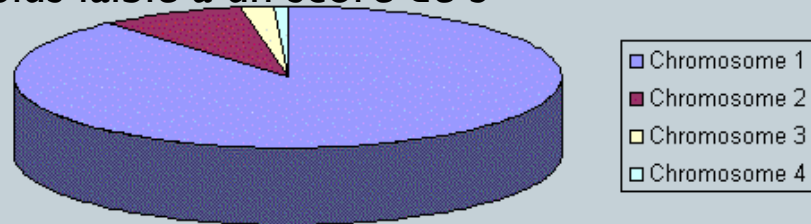
- Les parents sont d'abord triés par ordre d'adaptation (valeur de f), en commençant par les moins performants :

Le plus faible a un score de 1,

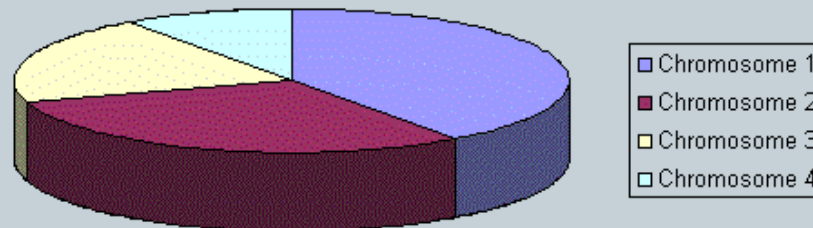
Le deuxième plus faible a un score de 2

Le troisième plus faible a un score de 3

Etc.



- On divise chaque score par la somme de tous les scores:



- On procède ensuite comme pour la roulette ; la méthode donne plus de chance aux chromosomes les plus faibles d'être repêchés

Sélection: Méthode du tournoi

28

- Partant de la population de n chromosomes, on forme n paires au hasard et on détermine le vainqueur dans chacune par sa valeur de f .
- Dans les paramètres de l'AG, on détermine une probabilité de victoire du plus chromosome le plus fort, représentant sa chance d'être sélectionné par la suite.
- Cette probabilité doit être grande (entre 70% et 100%).
- A partir des n paires, on sélectionne ainsi n individus pour la reproduction.

- Le croisement

Le croisement (crossover)



Croisement (Recombinaison)

- Deux individus (parents) échangent une partie de leurs chromosomes pour créer un ou plusieurs nouveaux individus (descendants). Les principales méthodes de croisement sont :
- **Croisement à un point** : Un point de coupure est choisi et les segments des parents sont échangés.
- **Croisement à deux points** : Deux points de coupure sont choisis et le segment intermédiaire est échangé.

Le croisement

31

Espérance d'amélioration de nouvelles générations

- Croisement à 1 point

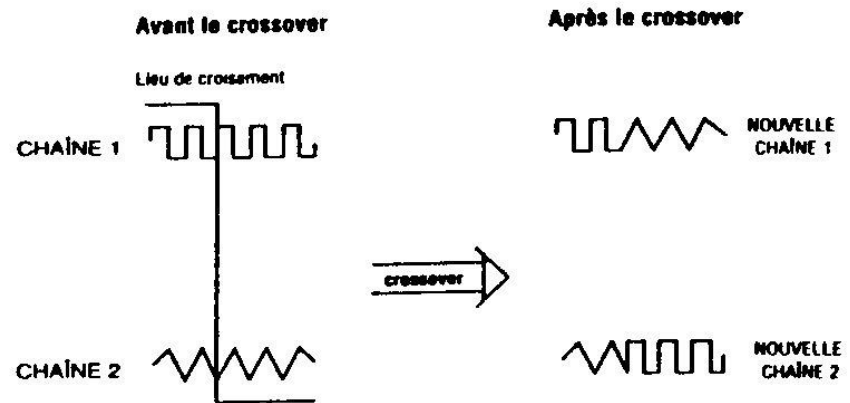
2 chromosomes de taille l .

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

exemple : $l=5$ et $k=3$



$$\begin{array}{l} C_1 = 0 \ 1 \ 1 \mid 0 \ 1 \\ C_2 = 1 \ 1 \ 0 \mid 0 \ 0 \end{array}$$



$$\begin{array}{l} C_{1,2} = 0 \ 1 \ 1 \mid 0 \ 0 \\ C_{2,1} = 1 \ 1 \ 0 \mid 0 \ 1 \end{array}$$

Le croisement

32

- Croisement à 2 points

2 chromosomes de taille l .

On considère qu'ils forment chacun un anneau fermé et on choisit aléatoirement 2 entiers k_1 et k_2 compris entre 1 et $l-1$.

k_1 et k_2 représentent les **points de croisement** des deux chaînes

On crée 2 nouveaux individus en échangeant les caractères des chaînes initiales compris entre k_1+1 et k_2

exemple : $l=10$, $k_1=1$ et $k_2=8$

$C_1 = 0 \mid 1110011 \mid 11$		$C_{1,2} = 0 \mid 1110000 \mid 11$
$C_2 = 1 \mid 1110000 \mid 00$		$C_{2,1} = 1 \mid 1110011 \mid 00$

- La mutation

La Mutation



- Après le croisement, une **mutation** peut se produire aléatoirement sur certains gènes pour introduire de la diversité dans la population.
- Cela permet d'explorer des zones de l'espace de recherche qui n'ont pas été couvertes par le croisement.

La mutation

35

- Permet de sortir de minima locaux
- 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** (≈ 0.001)
 - ↳ pour chaque caractère des descendants, probabilité de 1/1000 qu'il mute
- Si la mutation joue un rôle secondaire (dû au taux faible), elle permet l'exploration de dimensions (éventuellement utiles), abandonnées (à tort) par le processus de sélection ou absentes de la population initiale.



- Le remplacement

Le remplacement et l'élitisme



- Les nouveaux individus remplacent généralement certains membres de la population existante.
- L'**élitisme** garantit que les meilleures solutions actuelles sont préservées d'une génération à l'autre.

Le Remplacement

38



On travaille à population **constante**

Plusieurs stratégies :

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

Quels individus conserver ?

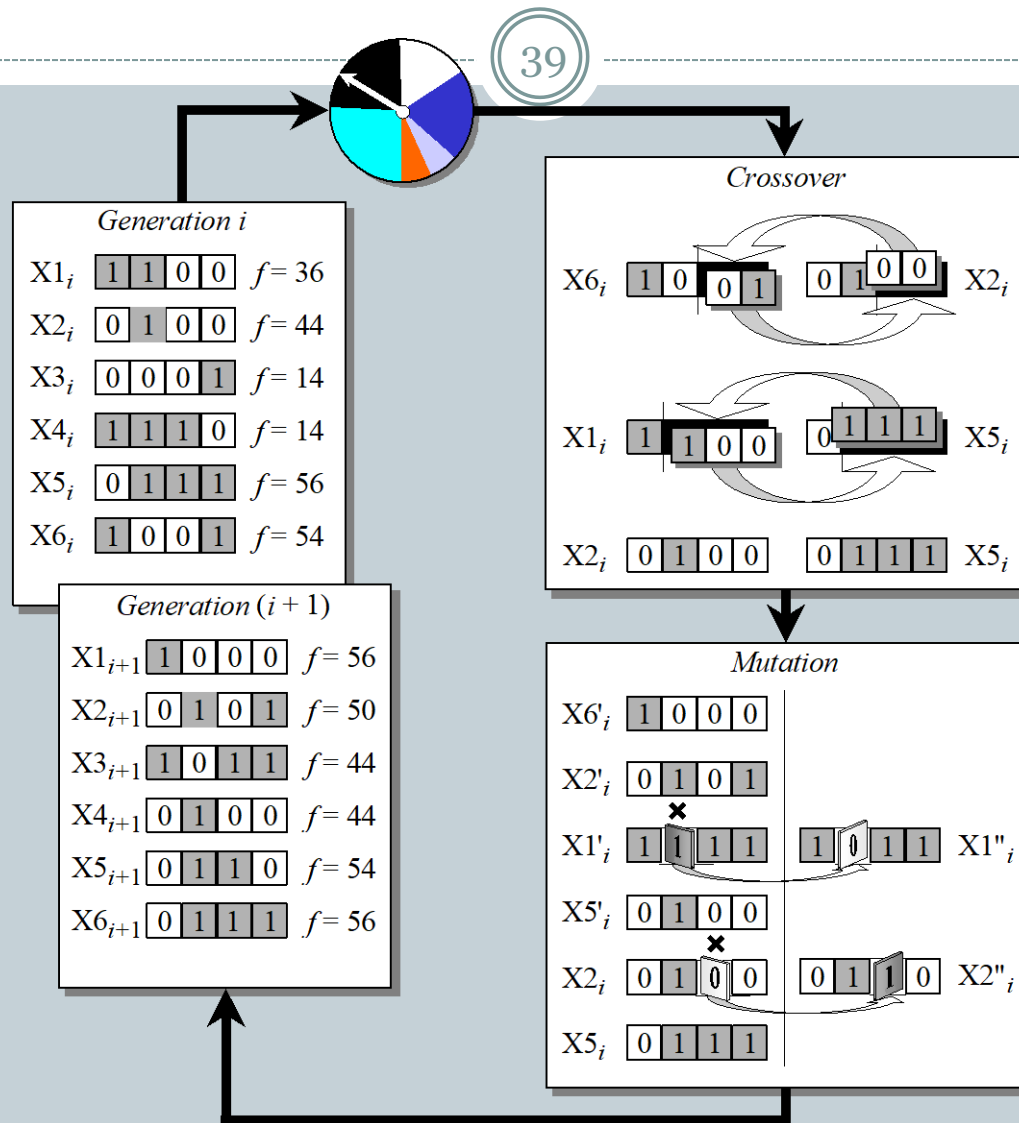
il est naturel de conserver les mieux adaptés :

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

- la stratégie **k-élitiste** : on garde systématiquement les k meilleurs individus d'une génération sur l'autre

Cycle de vie d'un algorithme génétique

39



Un exemple simple

40

- Nous reprenons ici l'exemple de Goldberg (1989).
- Trouver le maximum de la fonction $f(x) = x$ sur l'intervalle $[0,31]$ où x est un entier.
- La première étape consiste à coder la fonction.
- Par exemple, nous utilisons un codage binaire de x , la séquence (chromosome) contenant au maximum 5 bits.
- Ainsi, $x = 2 \rightarrow \{0,0,1,0\}$, de même $x = 31 \rightarrow \{1,1,1,1,1\}$.
- Nous recherchons donc le maximum d'une fonction de fitness dans un espace de 32 valeurs possibles de x .

Un exemple simple

41

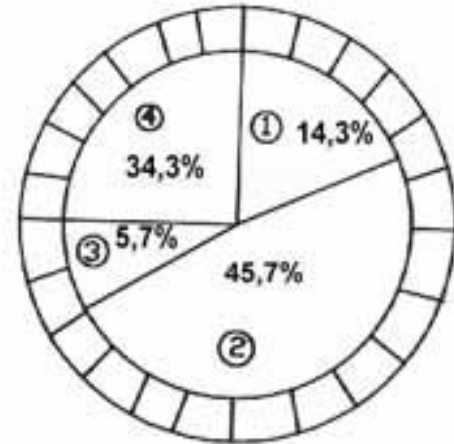
- **Tirage et évaluation de la population initiale**
- La taille de la population à $N = 4$.
- *Tirer donc de façon aléatoire 4 chromosomes sachant qu'un chromosome est composé de 5 bits, et chaque bit dispose d'une probabilité 12 d'avoir une valeur 0 ou 1.*
- Le maximum, 16, est atteint par la deuxième séquence.
- Voyons comment l'algorithme va tenter d'améliorer ce résultat.

Numéro	Séquence	Fitness	% du total
1	00101	5	14.3
2	10000	16	45.7
3	00010	2	5.7
4	00110	12	34.3
Total		35	100

Un exemple simple

42

- **Sélection**
- Une nouvelle population va être créée à partir de l'ancienne par le processus de sélection de la roue de loterie biaisée.
- Nous tournons cette roue 4 fois et nous obtenons au final la nouvelle population décrite dans le tableau.



Numéro	Séquence
1	10000
2	01100
3	00101
4	10000

Un exemple simple

43

- **Le croisement**
- Les parents sont sélectionnés au hasard.
- Nous tirons aléatoirement un lieu de croisement (site ou *locus*) dans la séquence.
- Le croisement s'opère alors à ce lieu avec une probabilité pc .

- *Les conséquences*

de cet opérateur en supposant que les chromosomes 1 et 3, puis 2 et 4 sont appariées et qu'à chaque fois le croisement s'opère (par exemple avec $pc = 1$).

$l = 3$	$l = 2$
100 00	01 100
001 01	10 000
10001	01000
00100	10100

Un exemple simple

44

- **La mutation**
- Dans cet exemple à codage binaire, la mutation est la modification aléatoire occasionnelle (de faible probabilité) de la valeur d'un bit (inversion d'un bit)
- Pour chaque bit est tiré un chiffre aléatoire entre 0 et 1 et si ce chiffre est inférieur à pm alors la mutation s'opère.
- $pm = 0:05$.

Anc. Chr.	Tirage aléat.	Nveau Bit	Nveau Chr.
10001	15 25 36 04 12	1	10011
00100	26 89 13 48 59	—	00100
01000	32 45 87 22 65	—	01000
10100	47 01 85 62 35	1	11100

Un exemple simple

45

- **Retour à la phase d'évaluation**
- Le maximum est maintenant de 28 (séquence 4).

Numéro	chaîne	Fitness	% du total
1	10011	19	32.2
2	00100	4	6.8
3	01000	8	13.5
4	11100	28	47.5
Total		59	100

- passé de 16 à 28 après une seule génération.
- Bien sûr, recommencer la procédure à partir de l'étape de sélection jusqu'à ce que le maximum global, 31, soit obtenu, ou bien qu'un critère d'arrêt ait été satisfait.

Pourquoi ça marche ?

46

un caractère
d'un individu

=

une
information

croisement = échange d'informations entre individus = formation de nouveaux "points de vue" (par destruction)

la mutation = dimension aléatoire

- éviter l'oubli de caractéristiques
- sortir d'un minimum local

reproduction
et
sélection



favoriser les informations présentes
dans les individus les mieux adaptés

Paramètres des Algorithmes Génétiques



- Les principaux paramètres influençant les performances d'un AG sont :
- **Taille de la population** : Nombre d'individus dans chaque génération.
- **Taux de croisement** : Proportion des individus qui se reproduisent.
- **Taux de mutation** : Fréquence des mutations.
- **Nombre de générations** : Nombre d'itérations de l'algorithme.
- **Critère d'arrêt** : L'algorithme s'arrête après un certain nombre de générations ou lorsque la solution a convergé.

Étapes de développement d'un algorithme génétique

48

1. Spécification du problème, définition des contraintes et des critères d'optimalité ;
2. Encodage du domaine du problème sous forme de chromosome
3. Définition de la fonction d'adaptabilité pour évaluer la performance du chromosome ;
4. Définition des opérateurs génétiques ;
5. Application de l'algorithme et réglage fin subséquents des paramètres.

Pour résumé

49

- chromosome = individu
- gène = partie constitutive d'un chromosome
- population = collection d'individus (la population est classée une fois que les individus ont pu être classés sur la base du critère donné)
- génération : une population est projetée sur plusieurs générations (la population initiale est générée de manière aléatoire)
- sélection, croisement, remplacement, mutations = opérateurs qui permettent de passer d'une génération à l'autre, tout en améliorant la population en base au critère donné.

Encodage

50

■ Codage binaire (données binaires)

○ Chromosome A 101100101100101011100101

Chromosome B 11111100000110000011111

Problème du sac à dos

■ Codage de permutation (données entières)

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

■ 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
- Structure d'arbre

Applications des algorithmes évolutionnaires



- **2.1. Optimisation combinatoire**

- Les AE sont souvent appliqués à des problèmes d'optimisation combinatoire, tels que : Problème du voyageur de commerce (TSP) : Trouver le chemin le plus court visitant un ensemble de villes.
- Planification et ordonnancement : Optimiser l'allocation des ressources dans divers contextes, comme la production ou la logistique.

- **2.2. Conception assistée par ordinateur**

- Dans le domaine de l'ingénierie, les AE sont utilisés pour : Conception de circuits électroniques : Optimiser les configurations pour améliorer la performance.
- Conception mécanique : Générer des formes innovantes qui répondent à des contraintes spécifiques.

- **2.3. Intelligence artificielle**

- **Les AE trouvent également leur place dans l'IA, notamment pour :**

- L'apprentissage automatique : Optimiser les hyperparamètres des modèles ou sélectionner les caractéristiques pertinentes.
- Jeux vidéo : Développer des agents autonomes capables d'apprendre et d'évoluer dans un environnement dynamique.

Conclusion



- Les algorithmes génétiques sont une méthode puissante et flexible pour résoudre des problèmes d'optimisation.
- Leur capacité à imiter les processus biologiques d'évolution les rend particulièrement efficaces pour explorer des espaces de recherche complexes, où d'autres méthodes échouent.
- Bien que leur convergence puisse parfois être lente et qu'ils soient sensibles aux paramètres, leur robustesse en fait un outil précieux dans de nombreux domaines scientifiques et industriels.