

Algorithmes génétiques

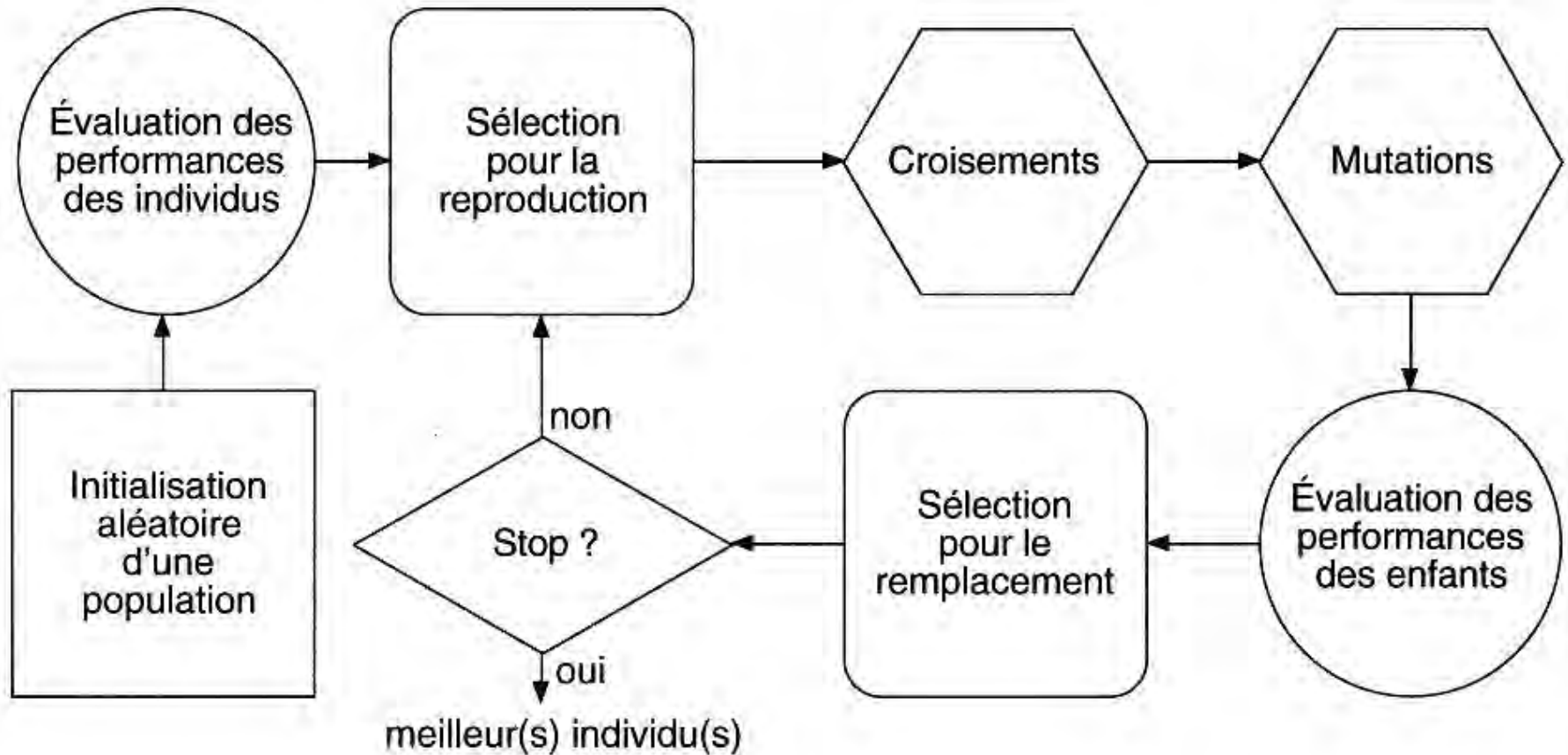


Algorithmes génétiques

- Inventés par Holland dans les années 70 [Holland, 75] pour imiter les phénomènes d'adaptation des êtres vivants.
- D'application plus générales, ils ont été adaptés aux problèmes d'optimisation par Goldberg [Golberg, 89].
- Elles connaissent depuis une quinzaine d'années un développement considérable grâce à l'augmentation de la puissance des calculateurs et l'apparition d'architectures parallèles à même d'exploiter leur « parallélisme intrinsèque ».

- Principe :
- On part d'une population de N solutions.
- On fait évoluer la population par croisement des individus les mieux adaptés et par quelques mutations.
- On garde la population constante en sélectionnant à chaque génération les plus adaptés.
- Est mieux adapté celui dont la fonction objectif prend les meilleures valeurs.

Organigramme d'un algorithme génétique



Paramètres d'algorithme génétique

- Codage de la population :

Un individu est une solution réalisable, ou *phénotype*. Il est codé par un vecteur de booléens, d'entiers, de 'float' (selon le problème), un *chromosome* ou *génotype*.

- fonction d'évaluation :

pour un problème d'optimisation c'est essentiellement la fonction objectif.

- Sélection des individus se reproduisant :

règle de sélection des individus se reproduisant à chaque génération.

- Opération de croisements :

règles de production des descendants.

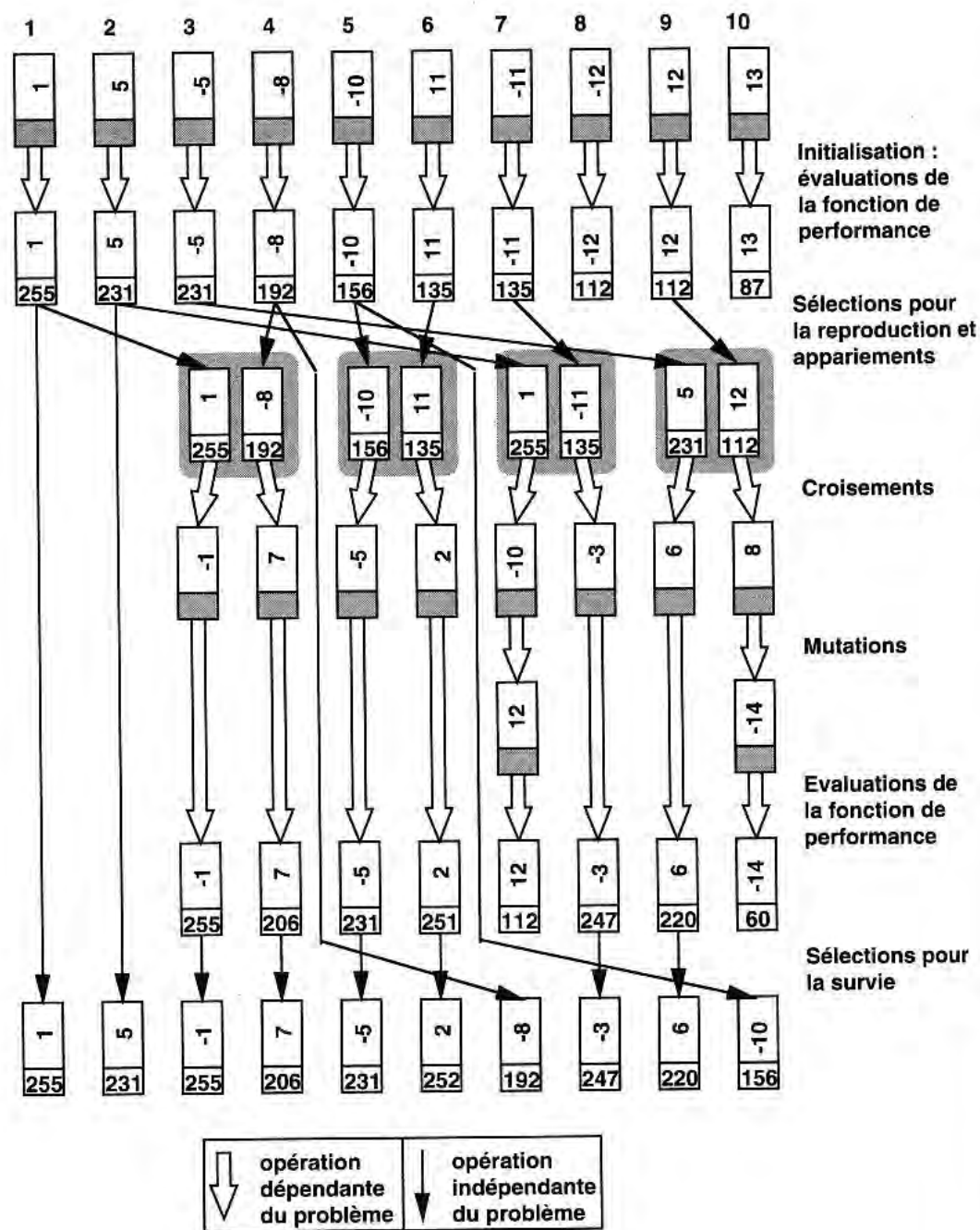
- Opération de mutation :

règle de mutation, probabilité.

- Sélection de la génération suivante :

parmi les individus parents+enfants, sélection de la génération suivante, en gardant la population constante.

- Exemple :
- On cherche à maximiser $C(x)=256-x^2$ sur les entiers dans l'intervalle $I=[-16,16]$.
- On se donne une population initiale de 10 individus tirés au sort dans I par une loi uniforme.
- On sélectionne les 8 meilleurs individus pour reproduction. On les apparie en couple par tirage au sort.
- Croisement : chaque couple $\{a,b\}$ produit 2 enfants obtenus par tirage au sort dans l'intervalle $[a,b]$.
- Mutation : avec une probabilité de 0.2 on change un individu par un tirage au sort dans I .
- On sélectionne ensuite les 10 meilleurs individus parmi les 10 (parents)+8 (enfants)=18 individus, pour obtenir la prochaine génération.



Sélection de $n \leq N$ parents parmi N individus

- La population est composée de N individus.
- On choisit n (pair) parents où $n \leq N$ est fixé.
- On constituera en suite $n/2$ couples, en général par tirage au sort, ou en couplant les meilleurs entre eux, ou le meilleur avec le pire, etc...

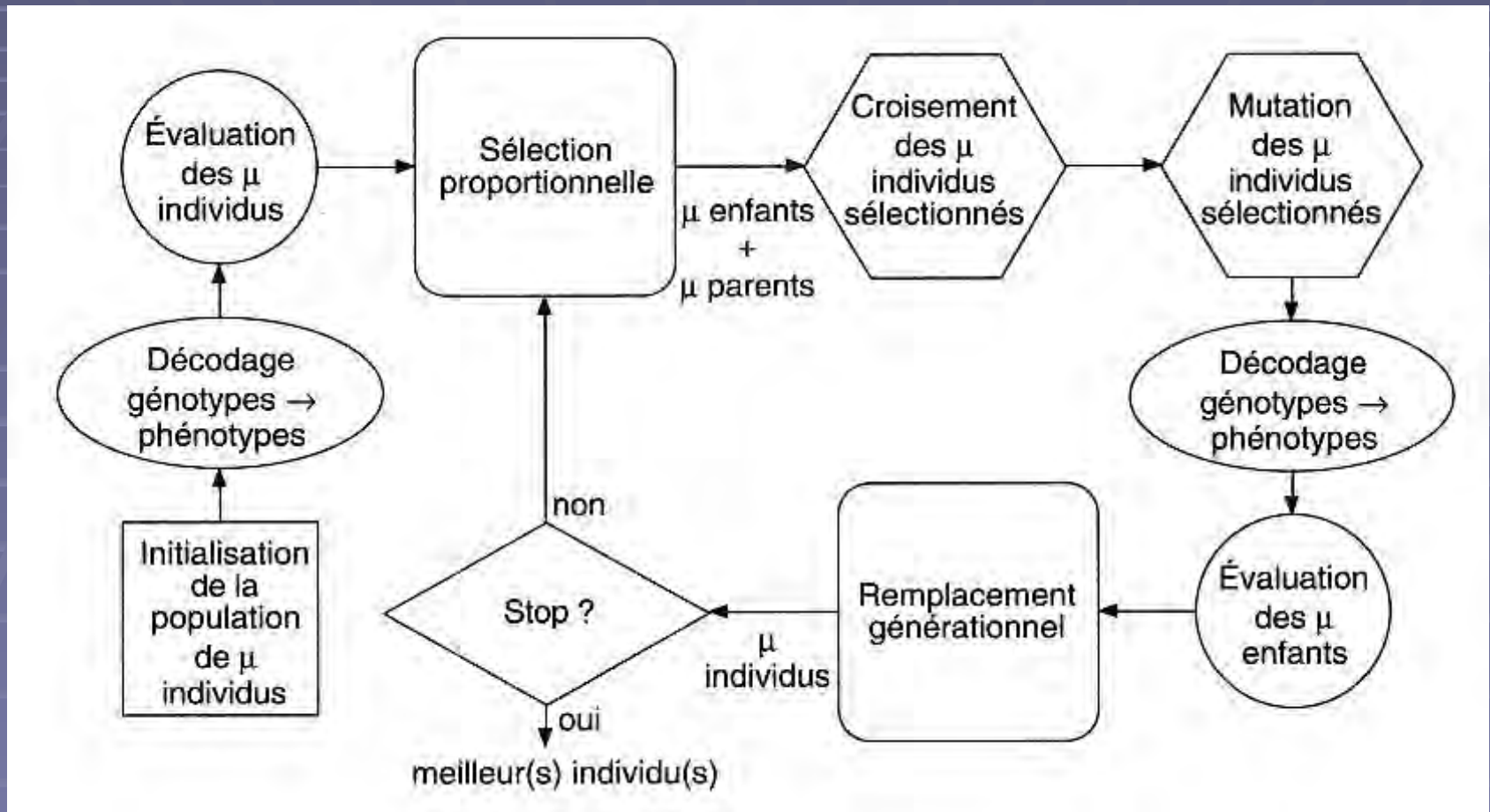
Sélection de $n \leq N$ parents parmi N individus

- **Sélection déterministe** : C'est la plus simple (à l'origine celle employée par tous les algo. génétiques). Elle choisit les n meilleurs. Elle fournit une faible diversité et peut rester bloquée dans des extrema locaux.
- **Sélection proportionnelle** : (ou par roulette). On tire au sort n individus. La probabilité pour chaque individu est proportionnelle à sa valeur pour la fonction f d'évaluation, i.e. $p(i) = n \times f(i) / (\sum_{k=1..N} f(k))$.
- **Sélection par tournoi déterministe** : Tirer au sort $k > 0$ individus et sélectionner le meilleur. Recommencer n fois (avec ou sans remise).
- **Sélection par tournoi stochastique** : Tirer au sort n couples. Dans chaque couple donner une probabilité $p \geq 0.5$ au meilleur des 2 (et $1-p$ à l'autre), puis tirer au sort dans chaque couple.

Sélection de la génération suivante

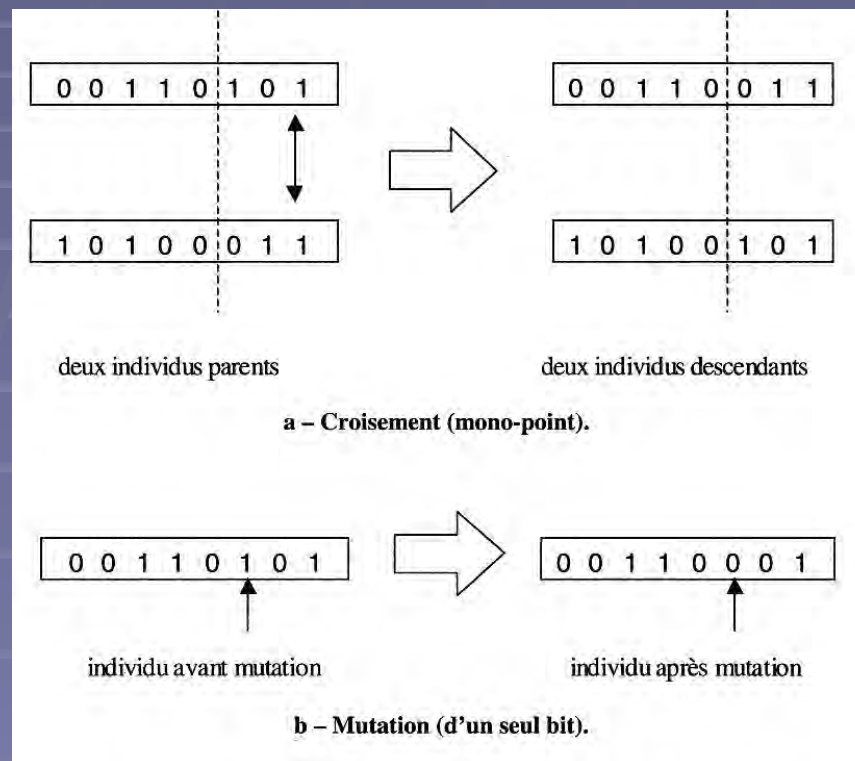
- **Remplacement générationnel** : c'est le plus simple (initialement le seul). On génère N enfants qui constitueront la nouvelle génération.
- **Remplacement par stratégie d'évolution** : On génère plus de N enfants et on garde les N meilleurs qui constitueront la prochaine génération.
- **Remplacement stationnaire** : la prochaine génération est constituée des N meilleurs parmi enfants+parents.
- **Remplacement élitiste** : A chaque génération on conservera k parents parmi les meilleurs que l'on complétera par les meilleurs enfants. Stratégie augmentant le taux de convergence mais diminue la diversité et donne plus de risque de rester piégé dans un extremum local.

Un schéma d'algorithme génétique simple



Croisement et mutation

- Les possibilités dépendent du génotype.
- La mutation s'effectue toujours avec une faible probabilité.
- Exemple pour un génotype binaire :

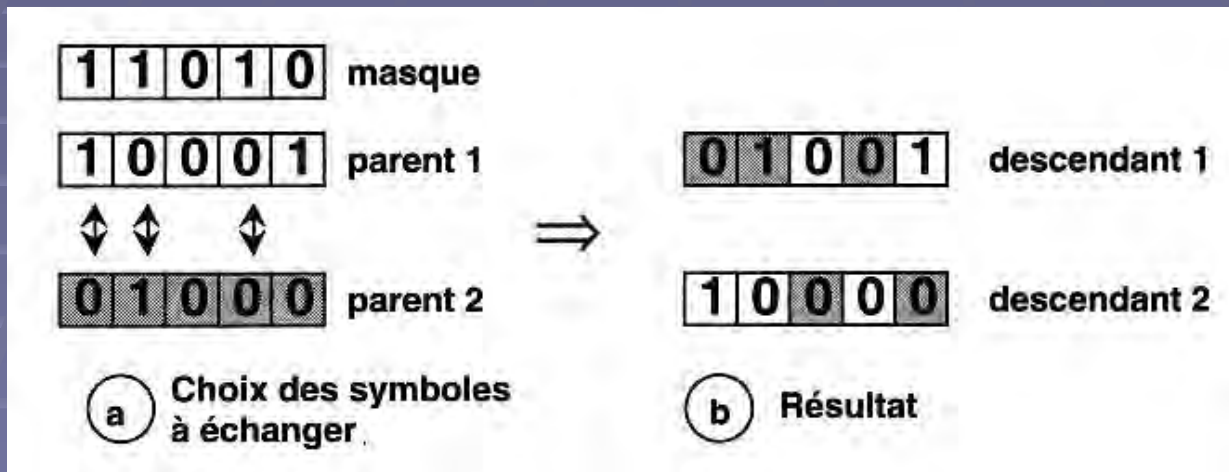


Croisement par
coupure '1 point'
(tirage au sort du
point de coupure)

Mutation par
modification
d'un bit tiré au
sort.

Croisement et mutation de génotypes binaires

- Le croisement de génotypes binaires peut se faire par coupure à '1 point', '2 points', ou plus. On obtient 2 descendants.
- Ou plus généralement par 'croisement uniforme' en tirant au sort un masque :



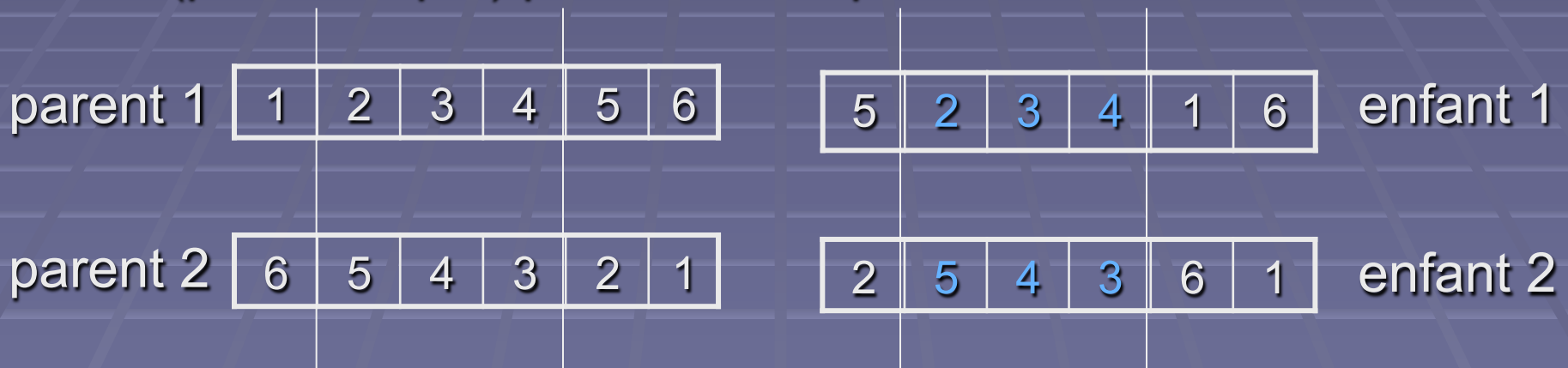
- La mutation peut se faire en inversant un ou plusieurs bits ou en permutant 2 ou plusieurs bits, tirés au sort.

Génotypes permutations d'entiers

- Pour le PVC ou un problème de coloration (par le biais d'une heuristique séquentielle), une solution réalisable est une permutations de n objets (les sommets du graphe).
- Un génotype pourra être représenté par n entiers de 1 à n sans répétition.
- Croisements et mutation doivent produire aussi une permutation.

Croisement de génotype permutation

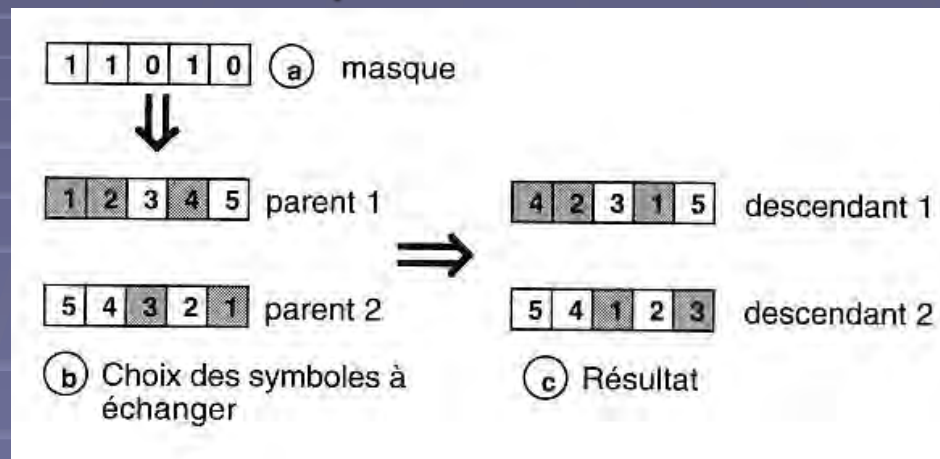
- Un croisement peut se faire par 'coupure' : tirer au sort 2 (par exemple) points de coupure et :



- On copie dans le 1er enfant un segment sur 2 du 1er parent que l'on complète par les nombres manquants dans l'ordre où ils apparaissent chez le deuxième parent. On fait l'inverse pour le deuxième enfant.

Croisement de génotype permutation

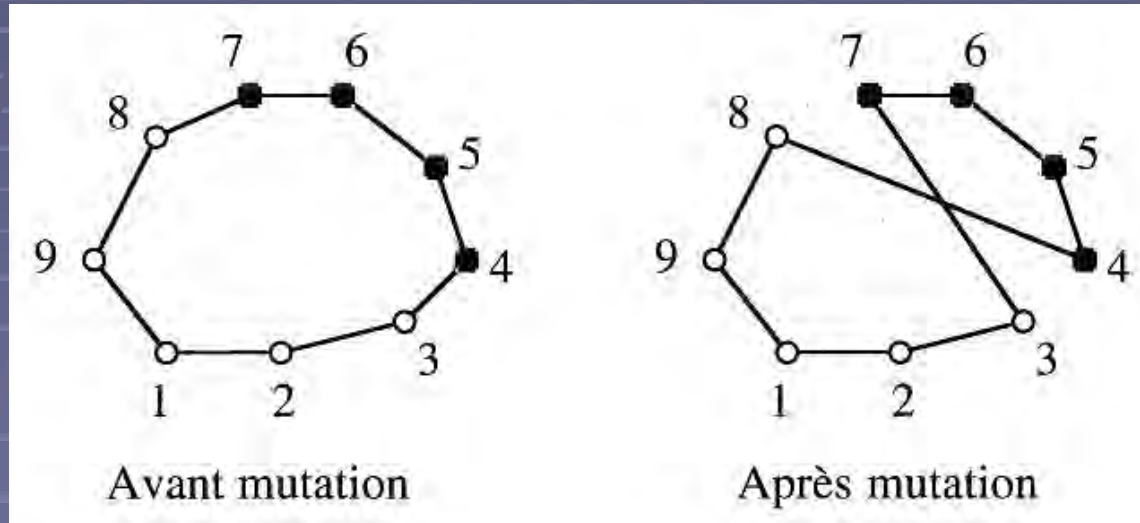
- Un croisement peut se faire par 'croisement uniforme' : on tire au sort un masque binaire :



- On copie dans le 1er enfant les nombres du 1er parent correspondant au masque que l'on complète par les nombres manquants dans l'ordre où ils apparaissent chez le deuxième parent. On fait l'inverse pour le deuxième enfant.

Mutation de génotypes permutation

- Une mutation 2-opt (par ex: pour le PVC) :



- Tirer au hasard 2 positions et inverser la chaîne qu'ils délimitent : $(1,2,3,\underline{4,5,6,7},8,9) \rightarrow (1,2,3,\underline{7,6,5,4},8,9)$.
- Mutation par permutation : changer au hasard la position d'un élément, ou appliquer une transposition (ou plus généralement une permutation) au hasard. (par ex: pour la coloration.)

Algorithmes génétiques :avantages

- Se prêtent aussi bien à l'optimisation combinatoire que continue, mais aussi à d'autres classes de problèmes (via le choix d'une fonction d'évaluation).
- Se prête très bien à l'optimisation multi-objectif, intervenant le plus souvent dans l'industrie.
- Si sa complexité est importante, son approche distributive (à chaque instant on travaille sur plusieurs solutions) se prête très bien aux calculs en parallèle, ce qui permet de contourner ce problème.
- Moderne, admet beaucoup de développements, dont la programmation génétique qui cherche le programme le mieux adapté à résoudre un problème.

Domaine d'application

- Donne de très bons résultats pour le PVC.
- Optimisation de la couverture d'un réseau radio mobile.
- Gestion du trafic aérien.
- etc...