



Algorithmes Génétiques (AG)

La Puissance des Algorithmes Génétiques dans la Résolution de
Problèmes Complexes

Présenté par:

- Essamih Hiba
- M'rhar Kaoutar
- Idahmed Mina

Plan

- Introduction
- Concepts Fondamentaux
- Opérateurs des algorithmes Génétiques
- Processus d'Algorithmes Génétiques
- Intensification et Diversification
- Applications
- Avantages et limitations
- Conclusion
- Références
- Implémentation

01

Introduction



Introduction

Les Algorithmes Génétiques (AG) ont la capacité de fournir une solution "suffisamment bonne" de manière "suffisamment rapide".

Le Piège des Minima Locaux

Les méthodes d'optimisation sans gradients, comme la recherche aléatoire ou la recherche par recuit simulé, peuvent avoir du mal à éviter les minima locaux lors de la recherche d'une solution optimale.

Lente Adaptation aux Changements Environnementaux

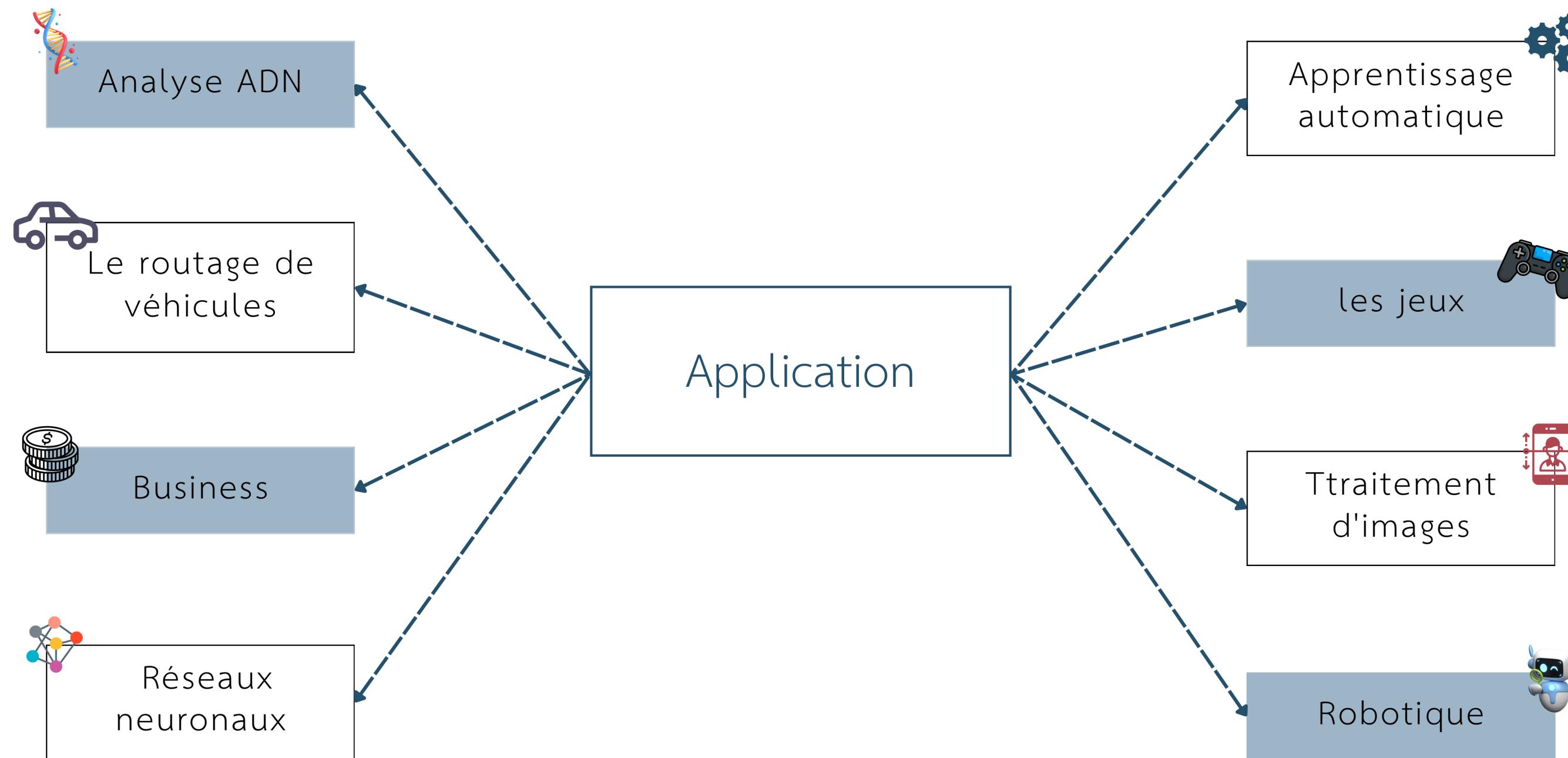
Certains algorithmes d'optimisation, comme les méthodes basées sur les gradients, peuvent avoir du mal à s'adapter rapidement aux changements dans l'environnement, tels que des variations soudaines dans la fonction objectif.

Limitations dans la Gestion des Solutions Discrètes et Combinatoires

Certains algorithmes d'optimisation sont moins efficaces pour traiter des espaces de solutions discrets et combinatoires, où les solutions sont représentées par des ensembles d'éléments discrets.

Introduction

les algorithmes génétiques sont largement utilisés dans différentes applications du monde réel



Introduction

- Les Algorithmes Génétiques ont été créés dans les années **1960** par **John Holland** pour modéliser les processus évolutifs en informatique. Sa publication majeure en **1975 "Adaptation in Natural and Artificial Systems"** a jeté les bases théoriques.
- Dans les années **1980**, les concepts ont été popularisés par Holland et ses collègues. Depuis lors, les Algorithmes Génétiques ont connu une croissance rapide, devenant des outils essentiels en informatique, intelligence artificielle et optimisation.



John Holland

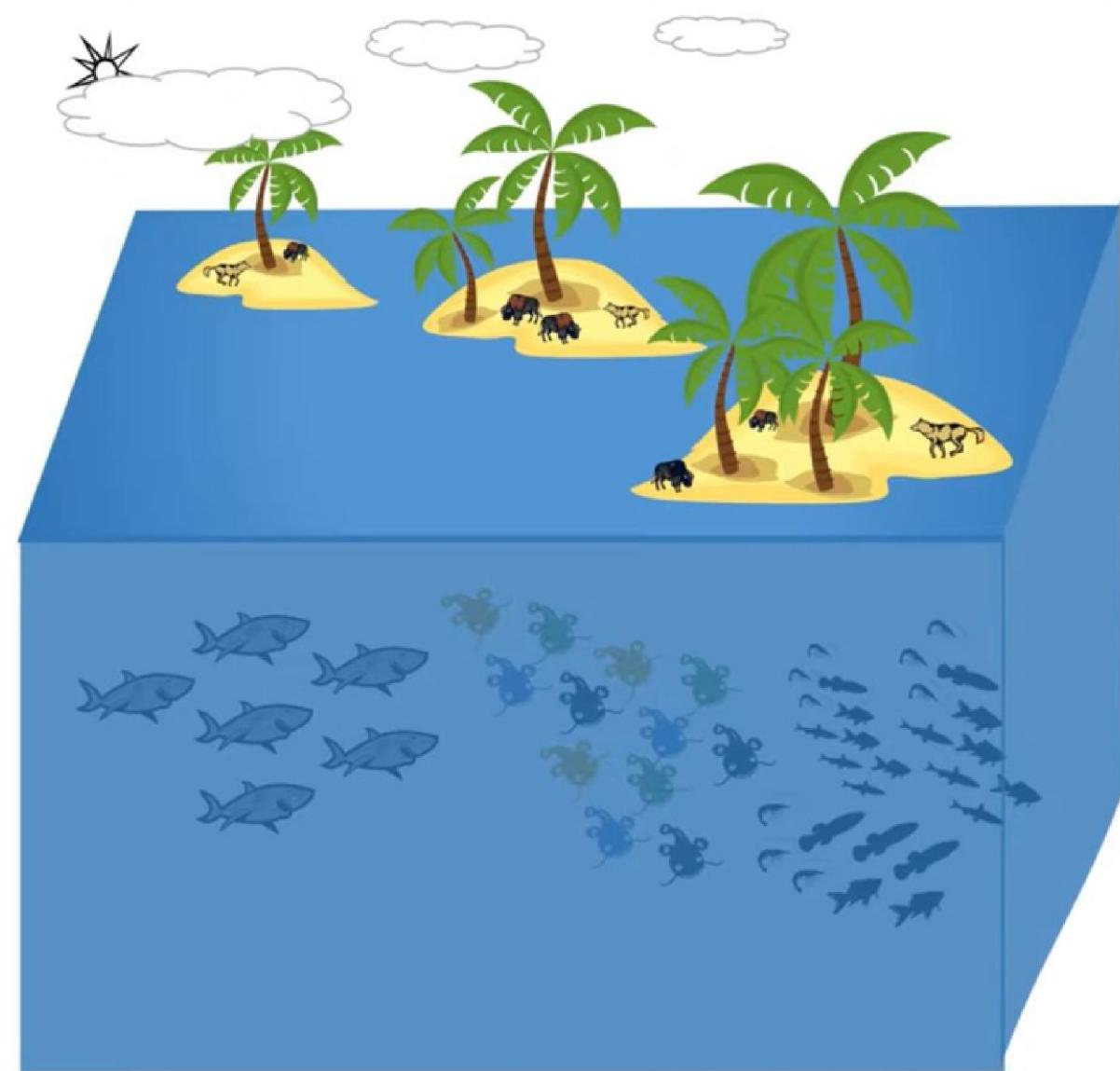
02

Concepts fondamentaux



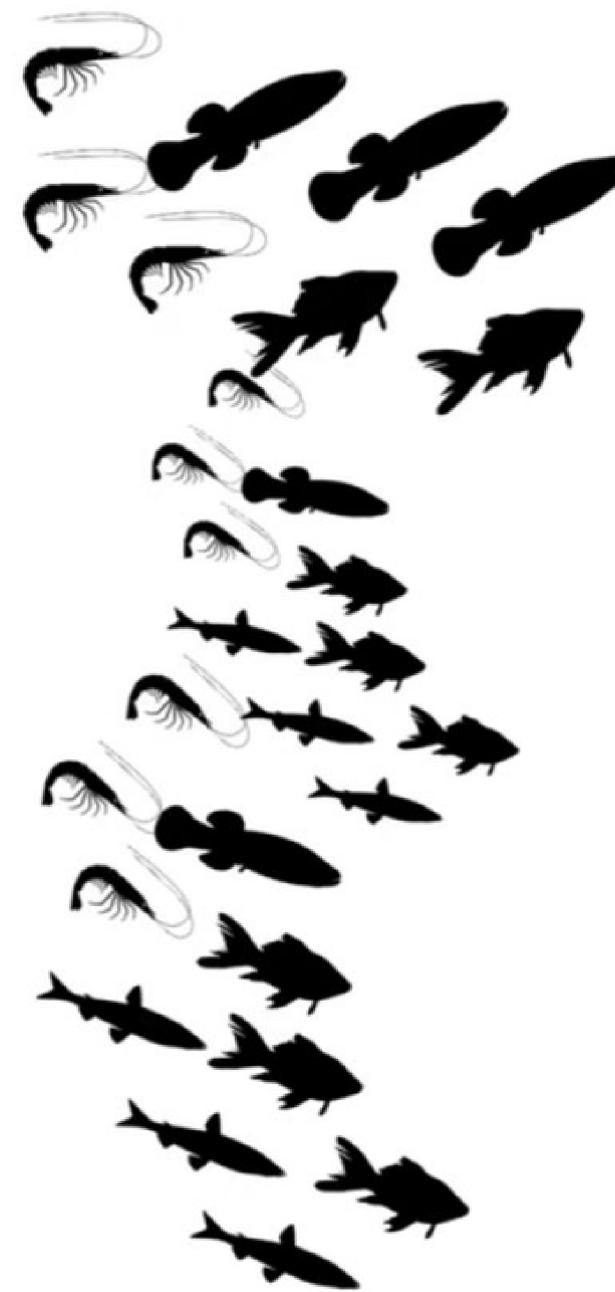
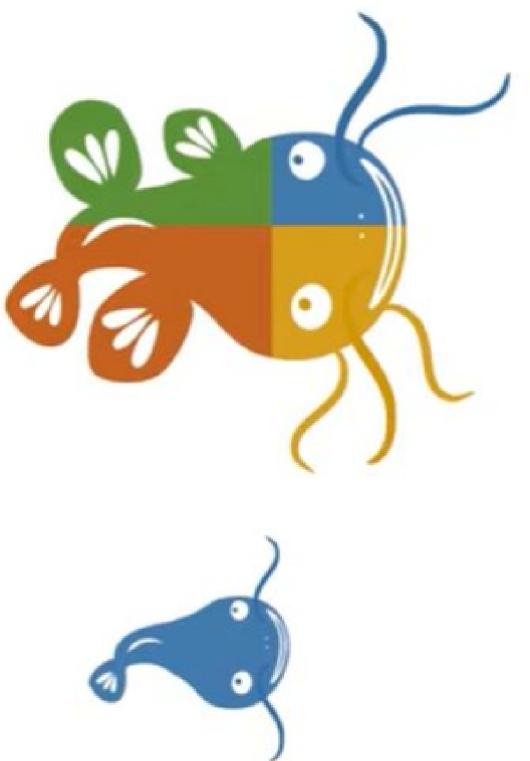
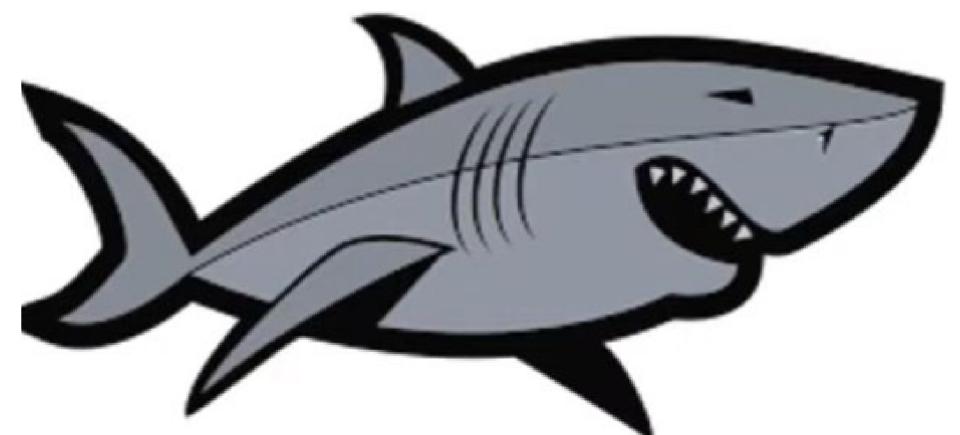
Inspiration

Les algorithmes génétiques sont un type d'algorithme d'optimisation inspiré par le processus de sélection naturelle et d'évolution.



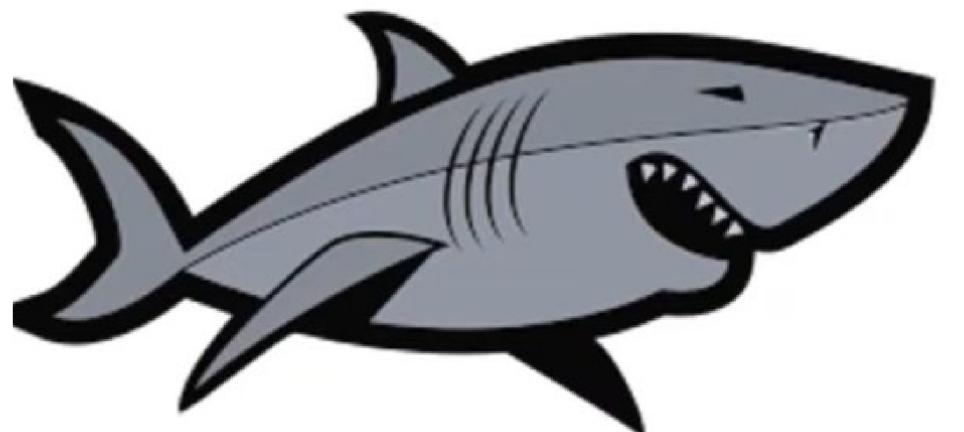
Inspiration

Selection

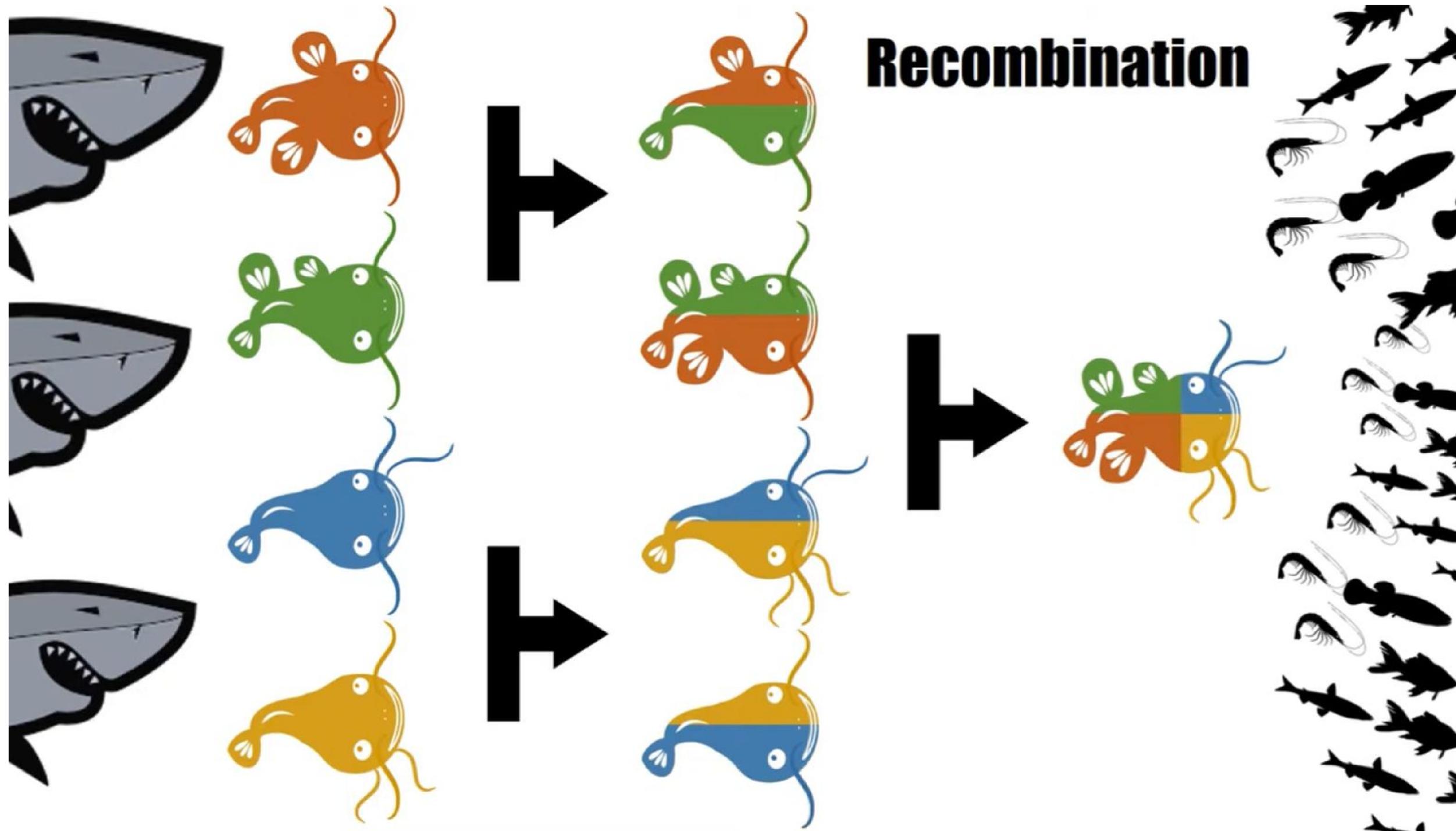


Inspiration

Selection

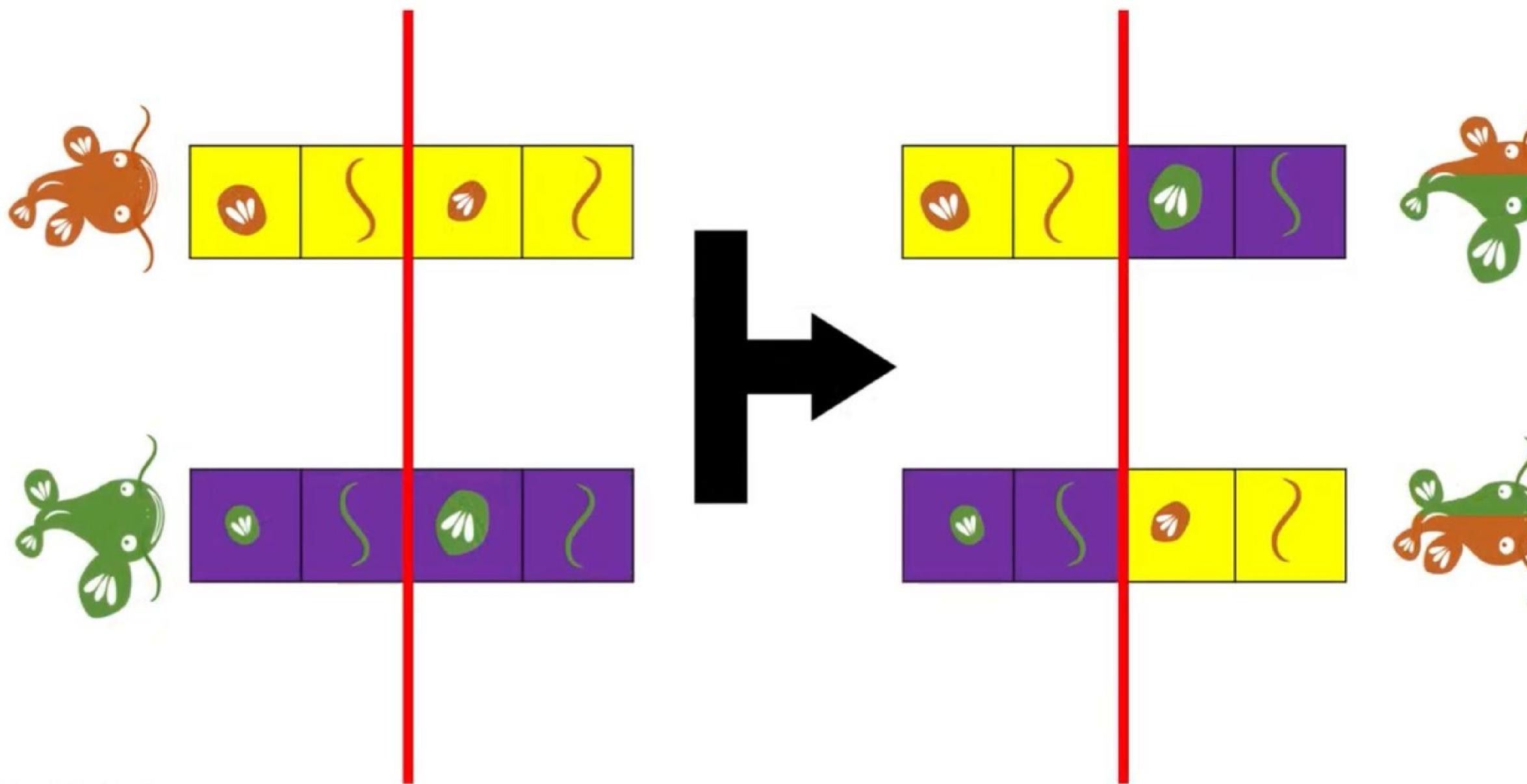


Inspiration

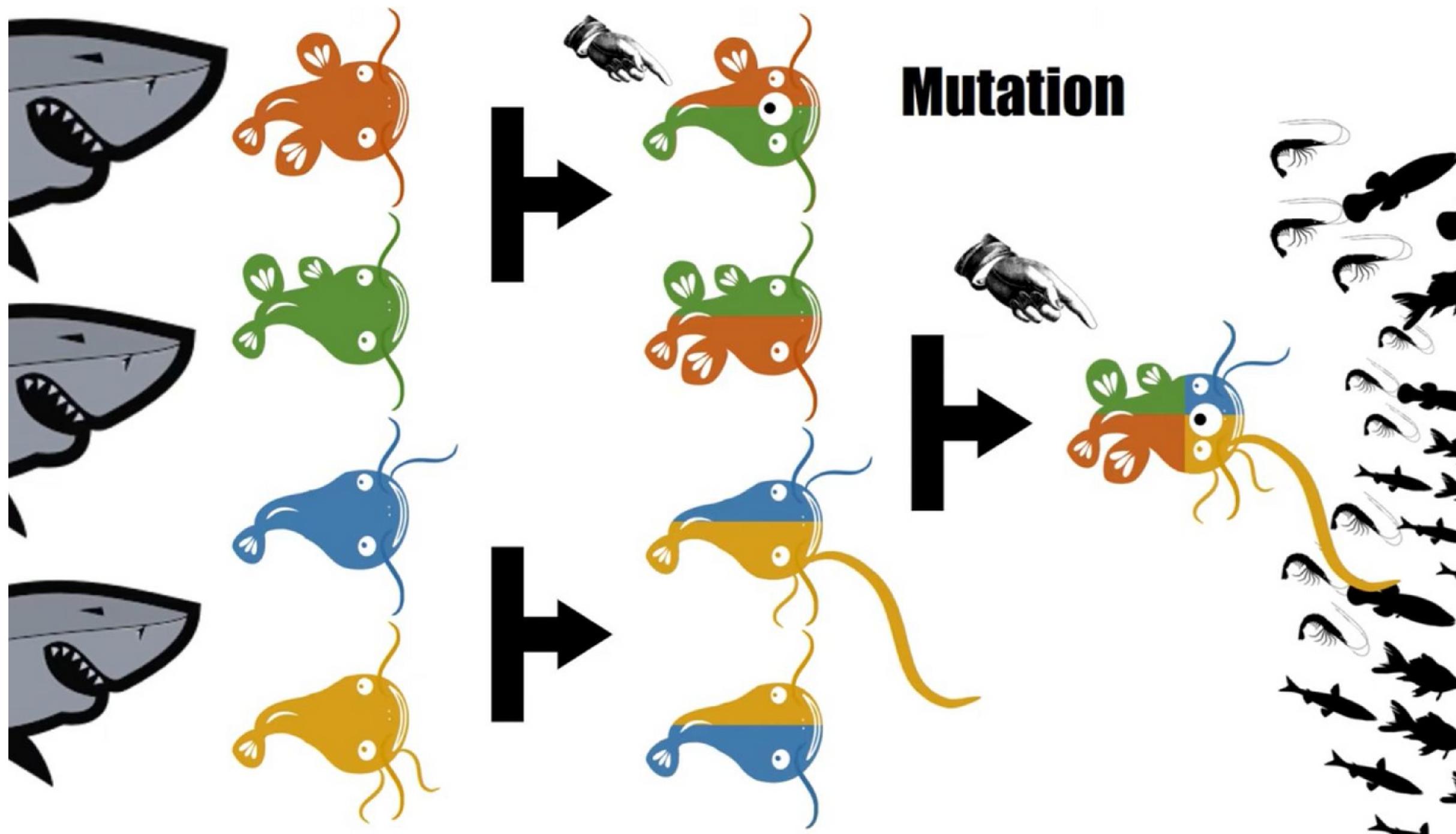


Inspiration

Recombination (crossover)

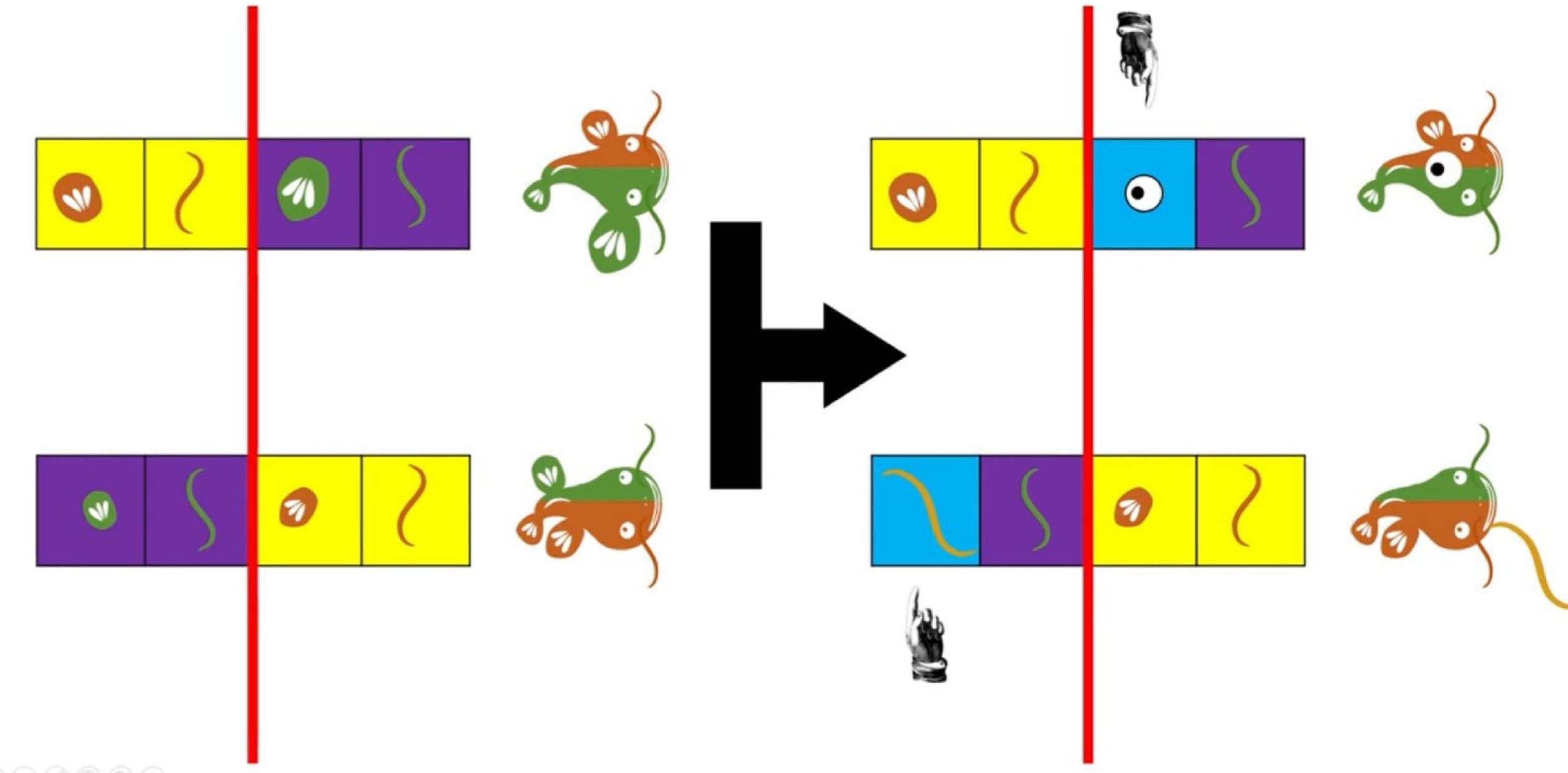


Inspiration



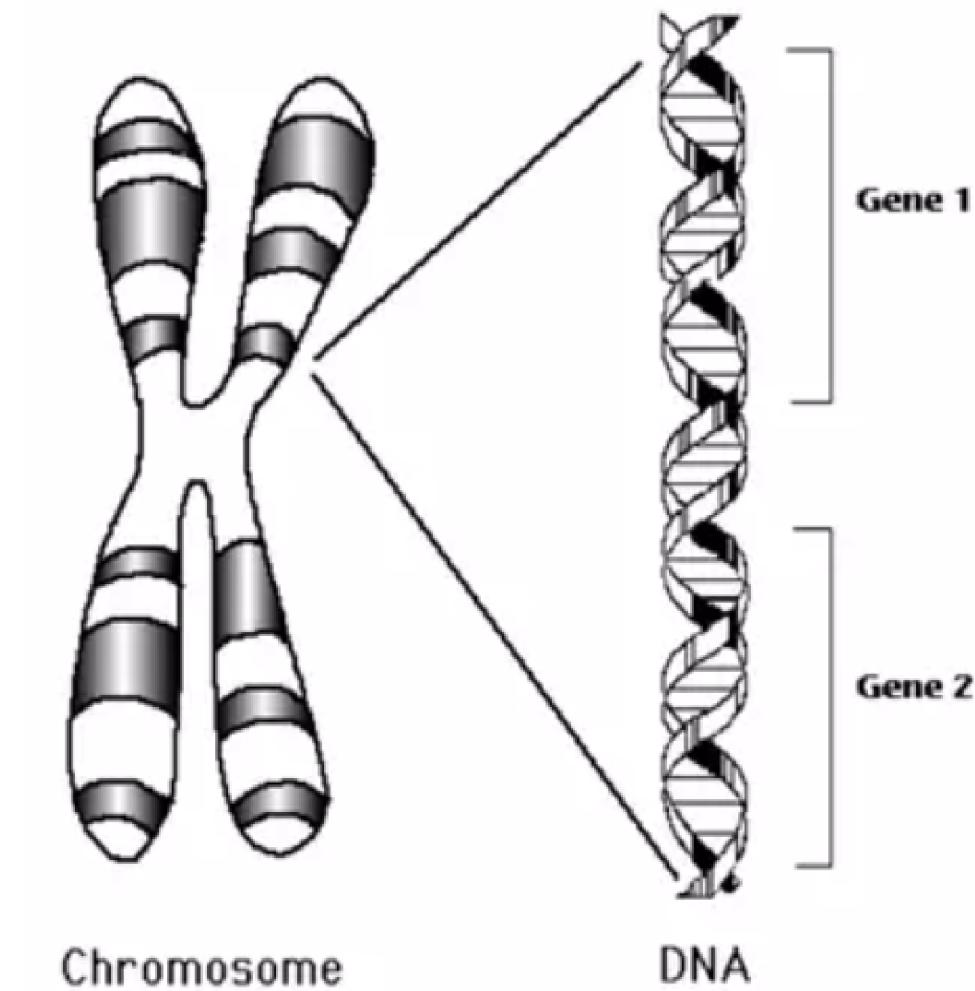
Inspiration

Mutation

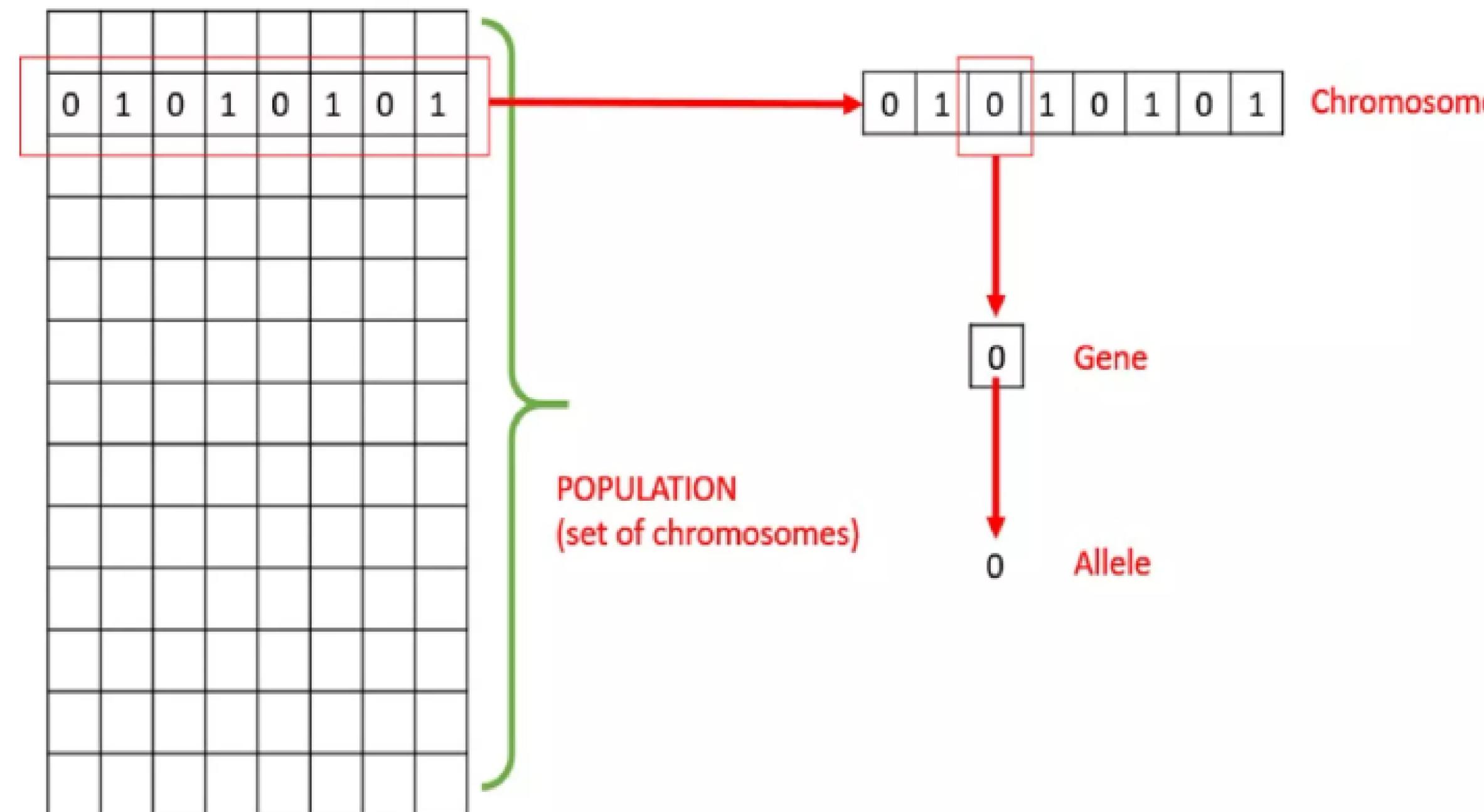


Les composants de base

- **Population** : C'est un sous-ensemble de toutes les solutions possibles pour un problème donné.
- **Chromosomes** : Un chromosome est une solution donnée à un problème.
- **Gène** : Un gène est une position élémentaire d'un chromosome.
- **Allèle** : C'est la valeur qu'un gène prend pour un chromosome particulier.



Les composants de base



03

Opérateurs des algorithmes Génétiques



Opérateurs des algorithmes Génétiques

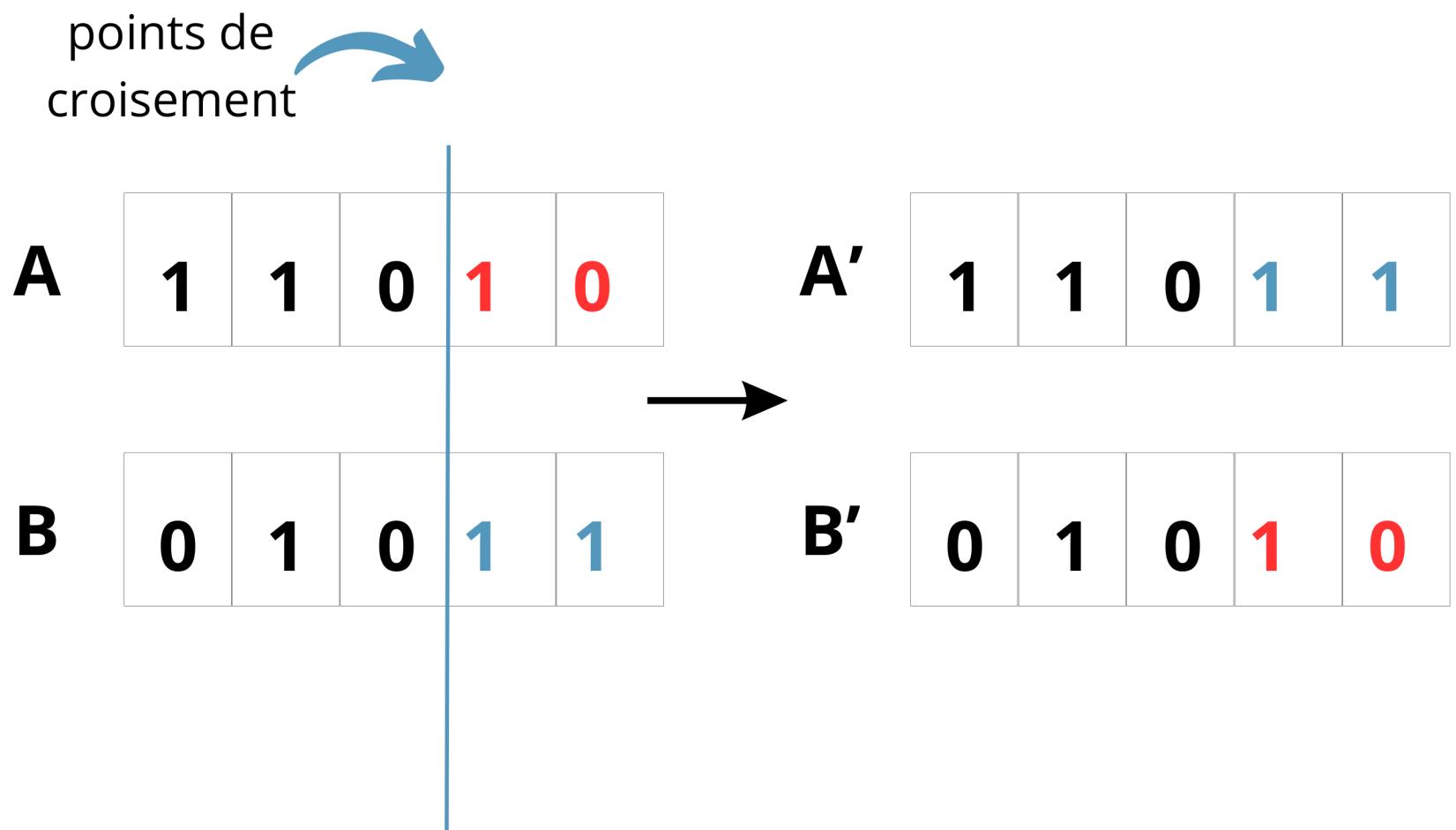
Les opérateurs génétiques jouent un rôle central dans les algorithmes génétiques, imitant les mécanismes de l'évolution naturelle.

Nous explorerons trois opérateurs clés qui sont cruciaux pour la performance et la diversité dans la recherche de solutions optimales :

- Crossover operation
- Mutation
- Parent Selection

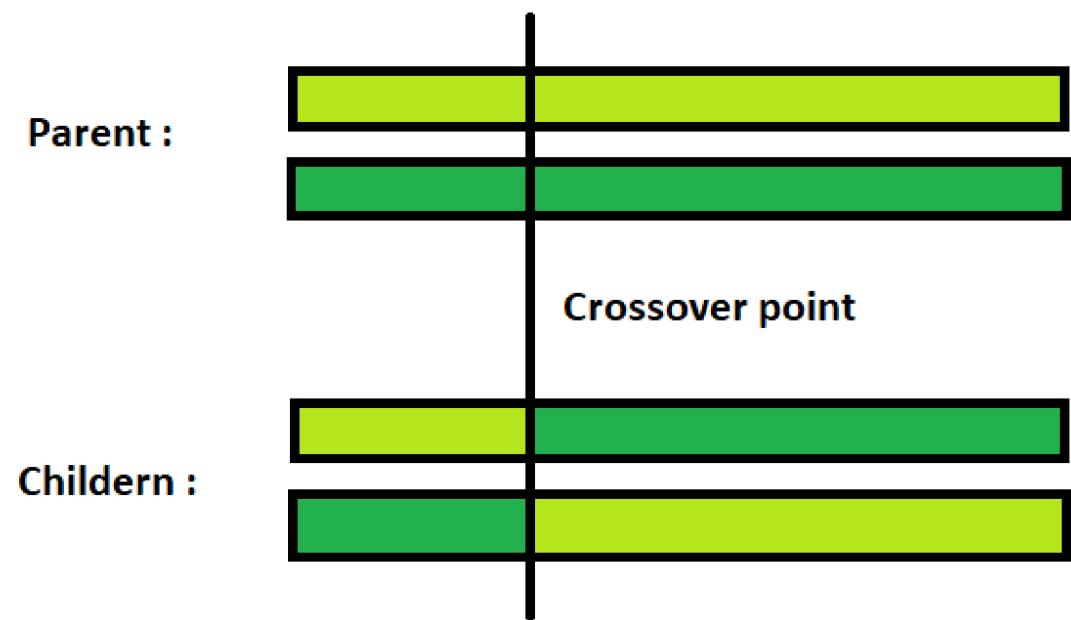
Crossover

- le croisement (crossover) est décrit comme un processus où les chromosomes parents échangent des segments génétiques. Des points de croisement sont définis, et les segments entre ces points sont échangés entre les chromosomes parents, créant ainsi des chromosomes descendants (OffSpring) qui combinent des caractéristiques de leurs parents respectifs.



Crossover

Single Point Crossover:

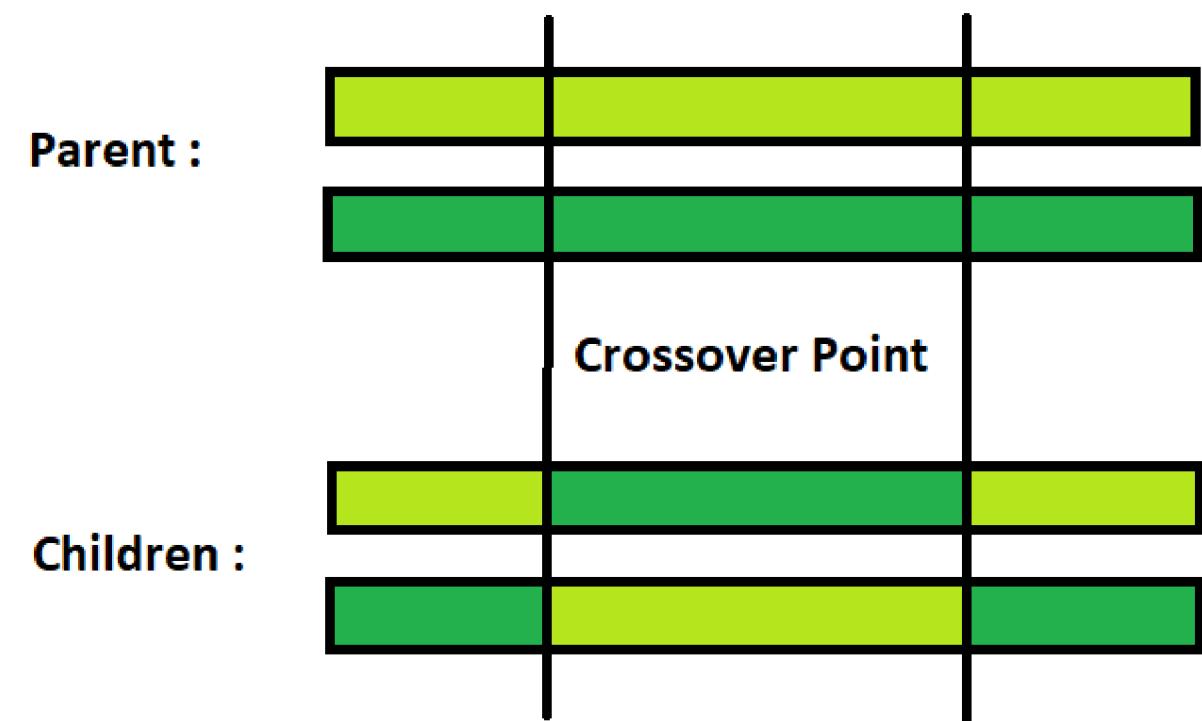


Chromosome1	11011 00100110110
Chromosome2	11011 11000011110
Offspring1	11011 11000011110
Offspring2	11011 00100110110

Single Point Crossover

Crossover

Two-Point Crossover:



Chromosome1	11011 00100 110110
Chromosome2	10101 11000 011110
Offspring1	11011 11000 110110
Offspring2	10101 00100 011110

Two Point Crossover

Crossover

Uniform Crossover:

Parent :

000000000000000000000000
1111111111111111111111

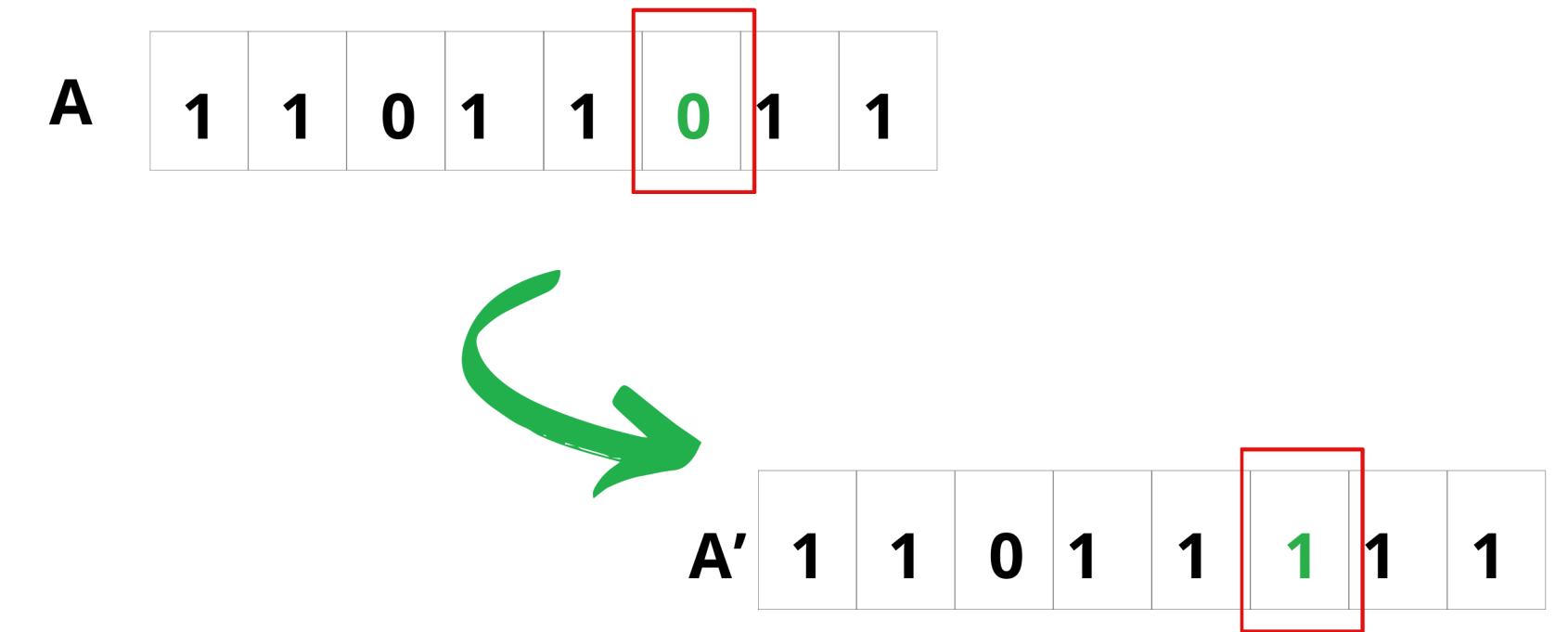
Children :

100011010100100111101
01110010110111000010

Uniform Crossover

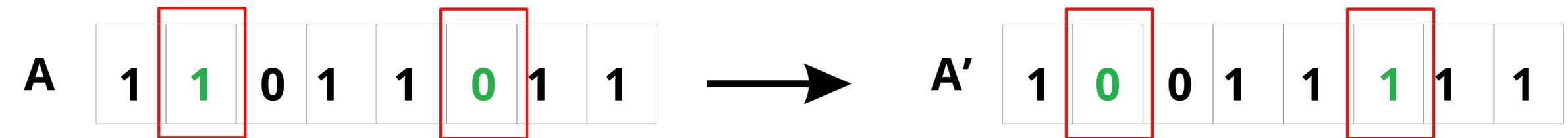
Mutation

- La mutation dans le cadre des algorithmes génétiques correspond à un processus où quelques gènes d'un chromosome sont changés de façon aléatoire. En d'autres termes, cette opération introduit des modifications aléatoires dans la séquence génétique d'un individu au sein d'une population.



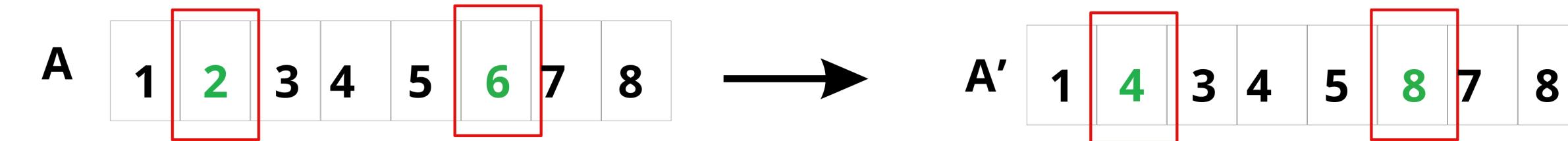
Mutation methodes

Bit Flip Mutation



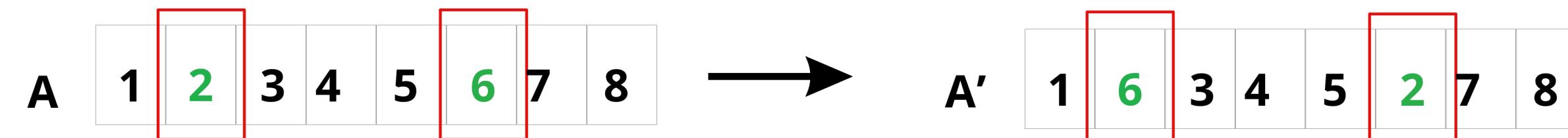
- introduit une petite perturbation locale dans le génome.

Random Resetting
Mutation



- introduit des variations aléatoires dans le génome.

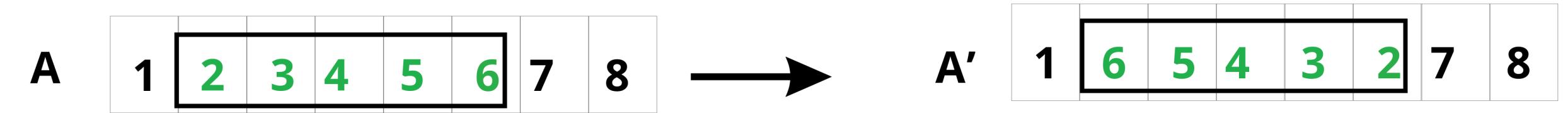
Swap Mutation



- déplacer l'information génétique entre différentes parties du génome .

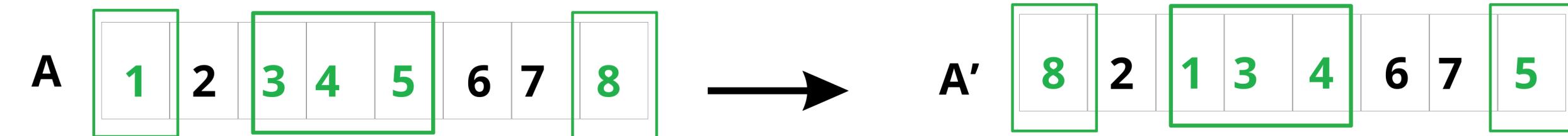
Mutation methodes

Inversion Mutation



- perturber l'ordre des gènes dans une portion du génome .

Scramble Mutation



- similaire a l'inversion perturbation dans une section du génome .

Parent selection

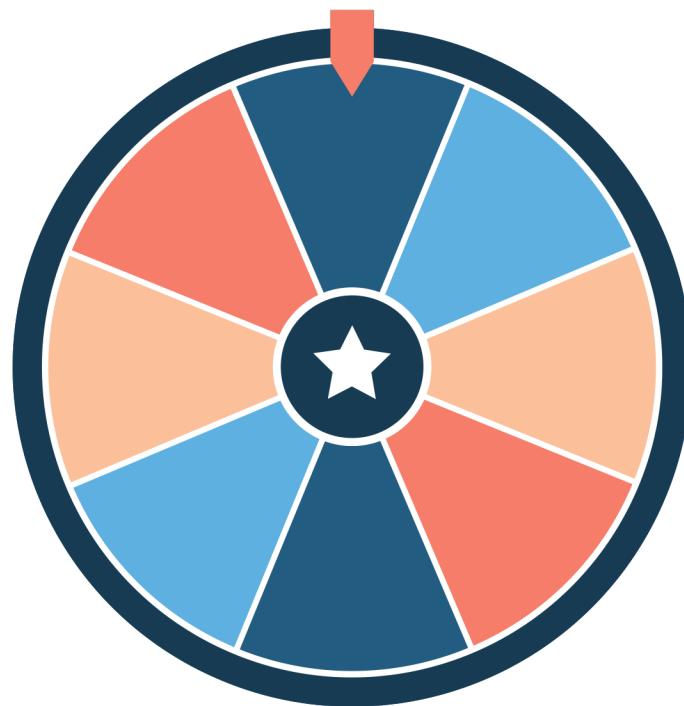
- La sélection des parents est une étape cruciale dans les algorithmes génétiques, où les individus de la population actuelle sont choisis pour produire une descendance.



Parent selection methodes

Roulette wheel selection

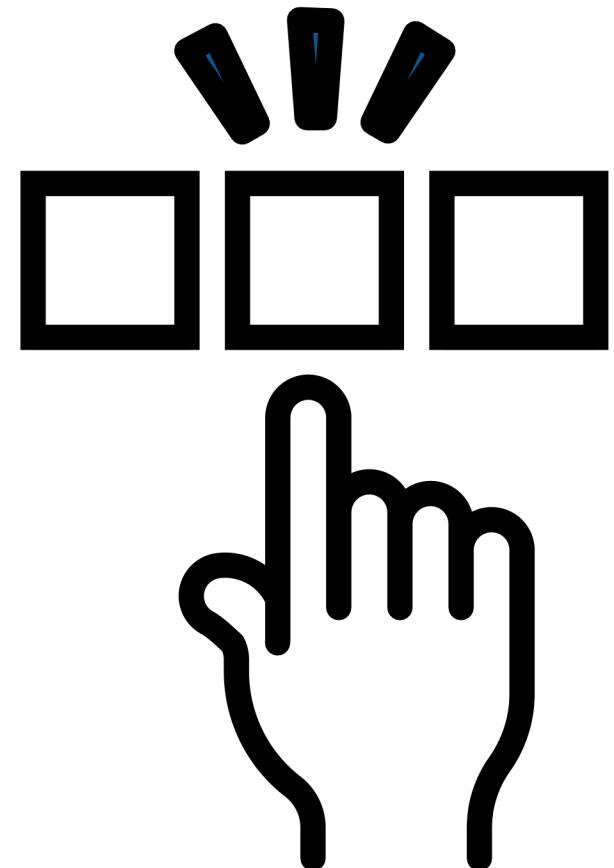
- Les individus sont sélectionnés avec une probabilité proportionnelle à leur fitness. Les individus plus aptes ont une plus grande chance d'être sélectionnés.



Parent selection methodes

Tournament Selection

- Dans la sélection par tournoi, chaque individu de la population est associé au hasard à un autre. Les valeurs de fitness de chaque paire sont comparées. L'individu le plus en forme de la paire avance vers la prochaine "manche", tandis que l'autre est éliminé. Ce processus se répète jusqu'à ce qu'il y ait un nombre de gagnants égal au nombre souhaité de parents.



Parent selection methodes

Elitism Selection

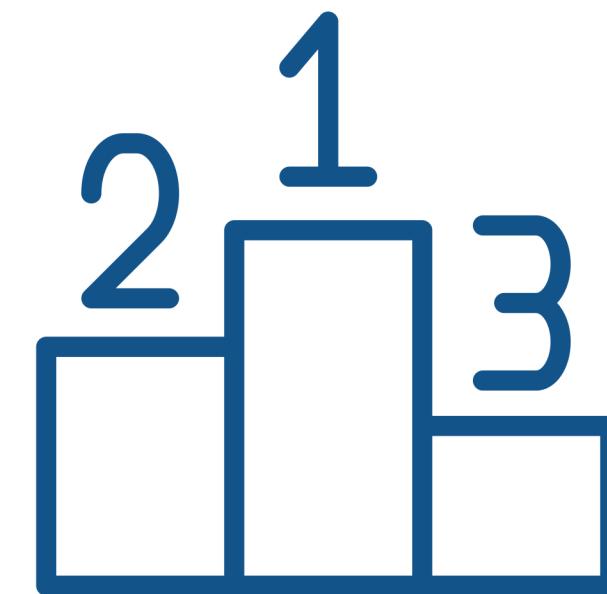
- la stratégie de sélection la plus simple et la plus courante dans les algorithmes génétiques. L'idée est simple : choisissez les n meilleurs chromosomes pour se reproduire. La sélection par élitisme donne la préférence aux chromosomes les plus performants



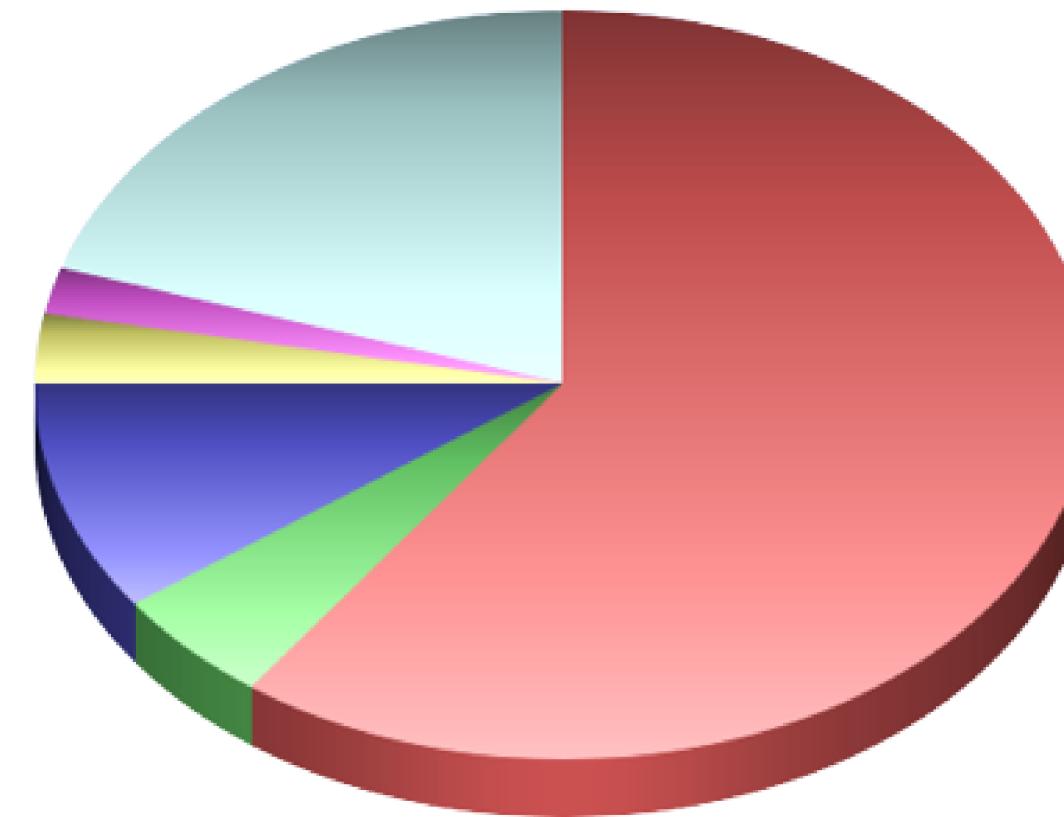
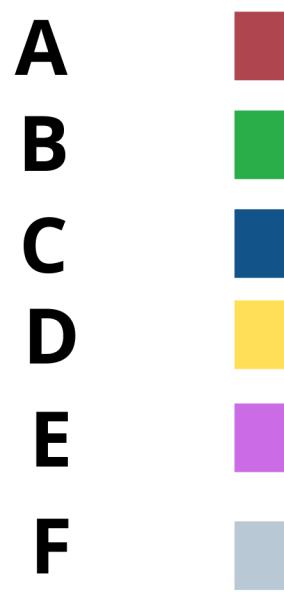
Parent selection methodes

Rank based selection

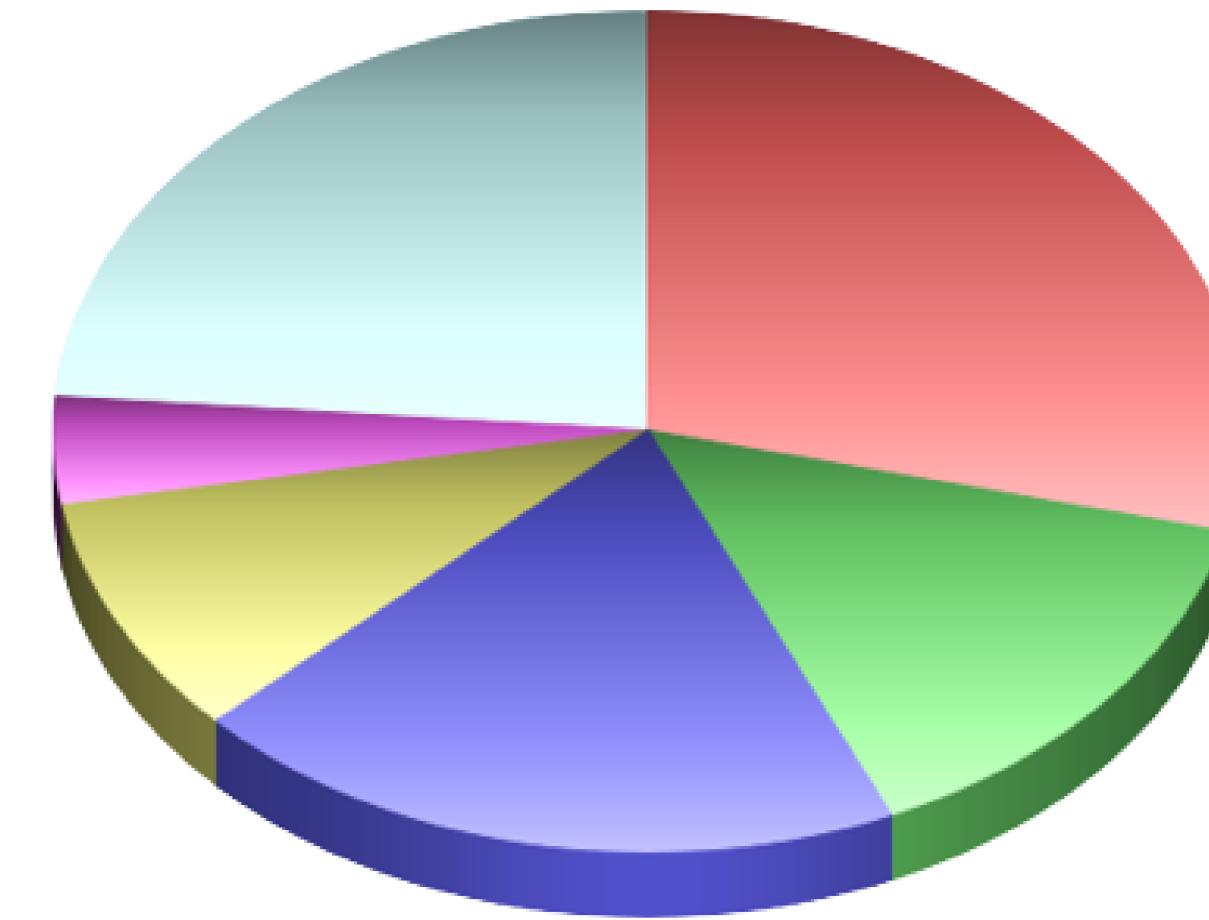
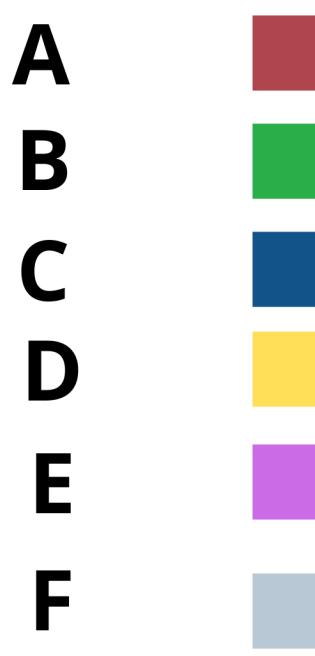
- La sélection par rang classe d'abord la population, puis chaque chromosome reçoit une aptitude en fonction de ce classement. Le moins performant aura une aptitude de 1, le deuxième moins performant 2, et ainsi de suite, jusqu'au meilleur qui aura une aptitude de N (le nombre de chromosomes dans la population). Après cela, tous les chromosomes ont une chance d'être sélectionnés.



Exemple :



Roulette wheel selection



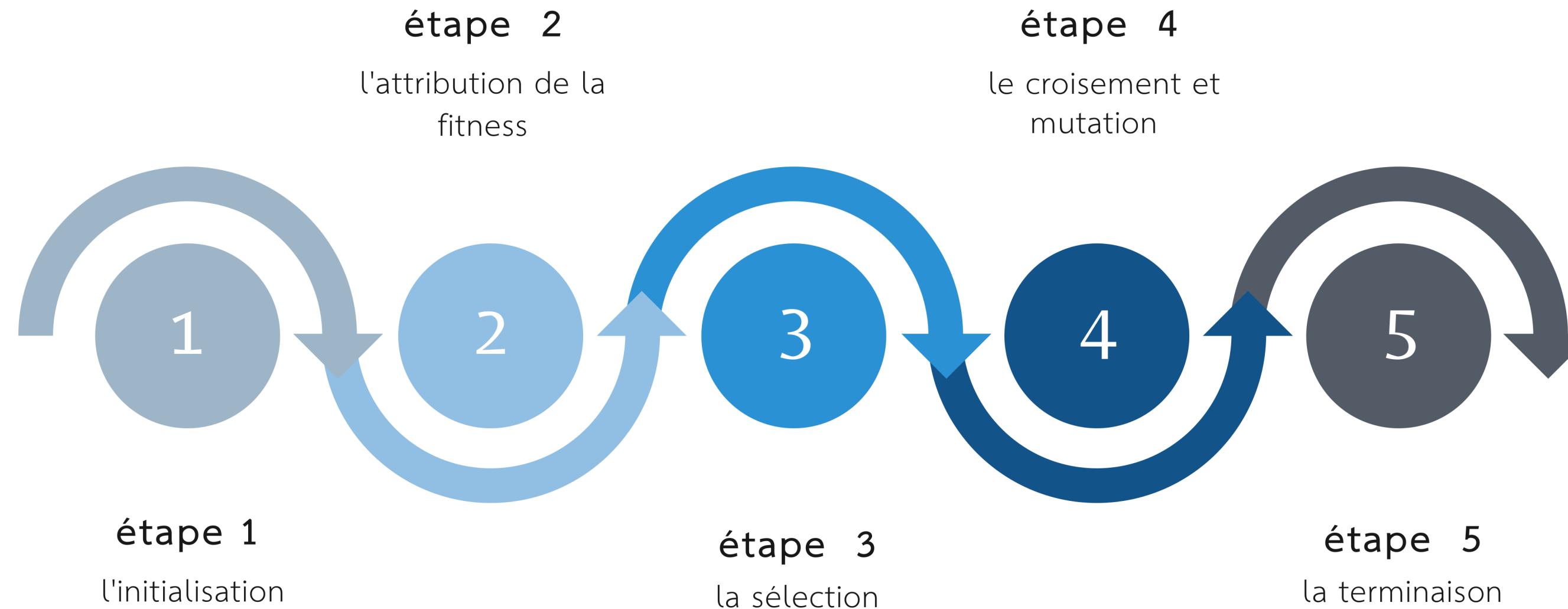
Rank base selection

04

Processus d'Algorithmes Génétiques



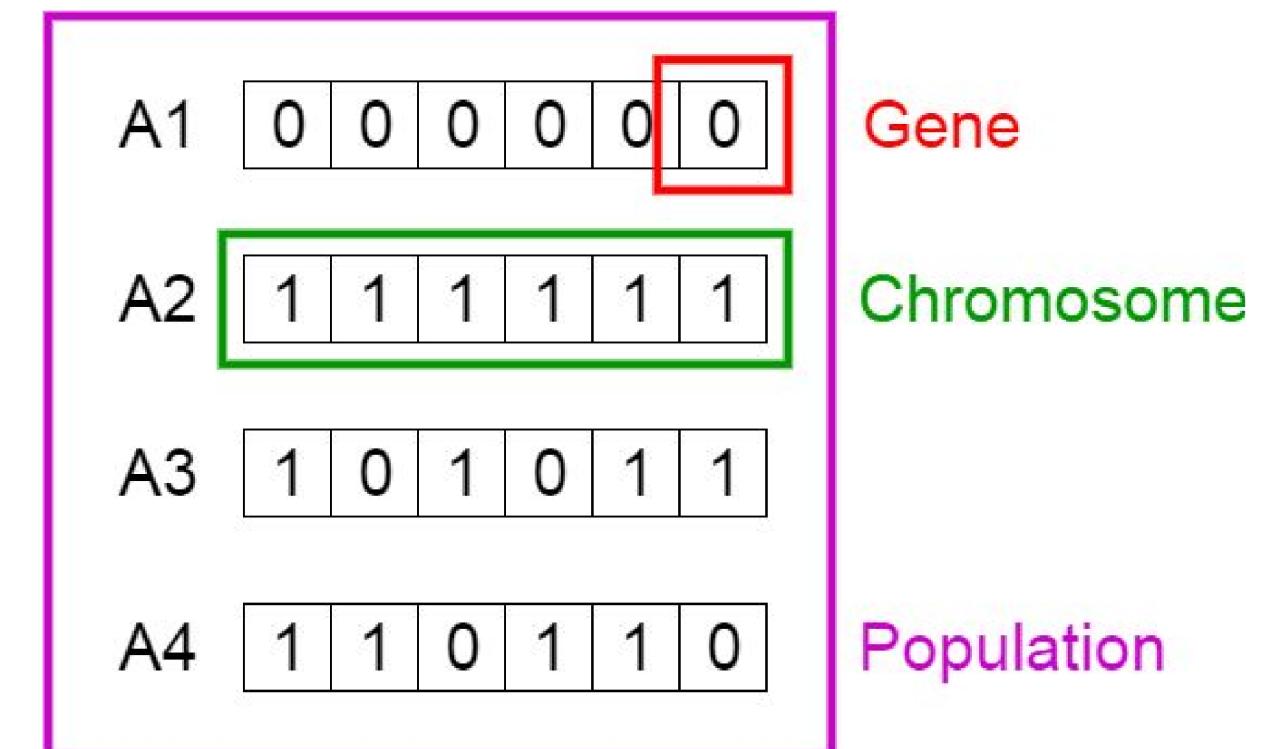
Processus d'Algorithmes Génétiques



Processus d'Algorithmes Génétiques

Population initiale:

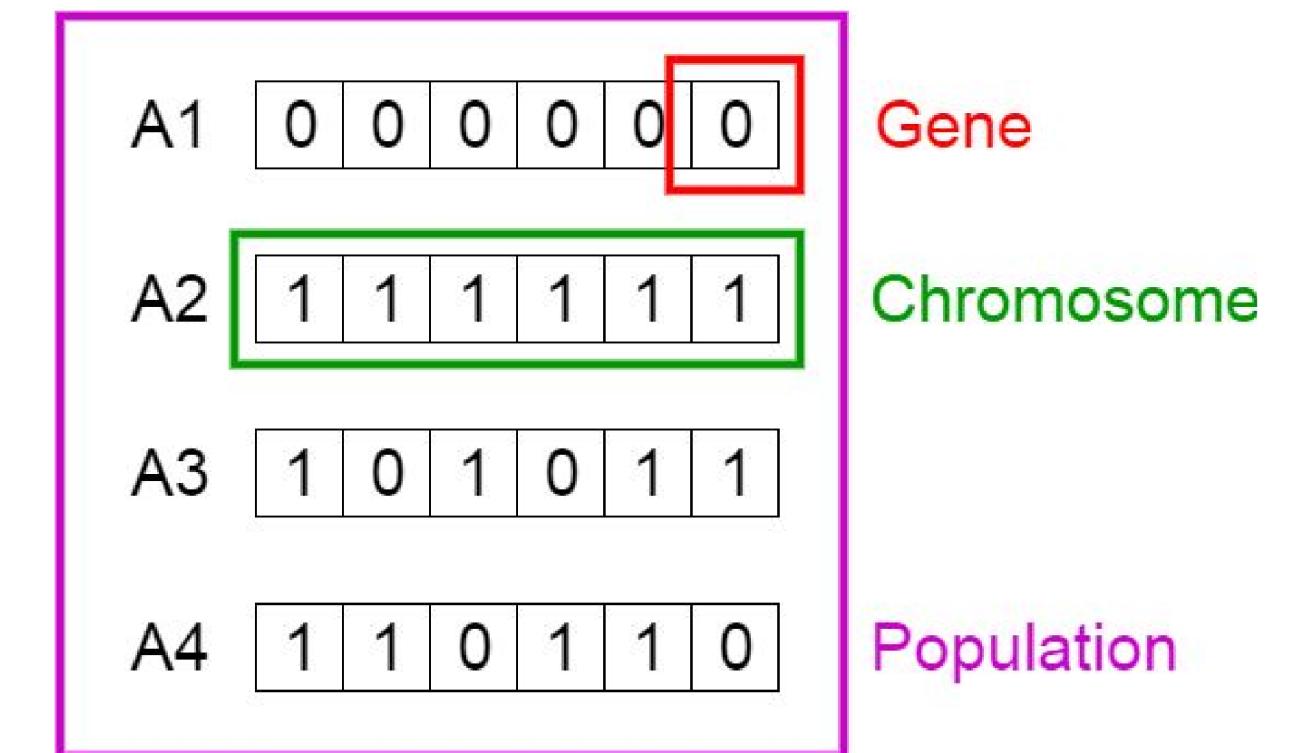
- Le processus débute avec un ensemble d'individus appelé une **Population**.



Processus d'Algorithmes Génétiques

La fonction de fitness:

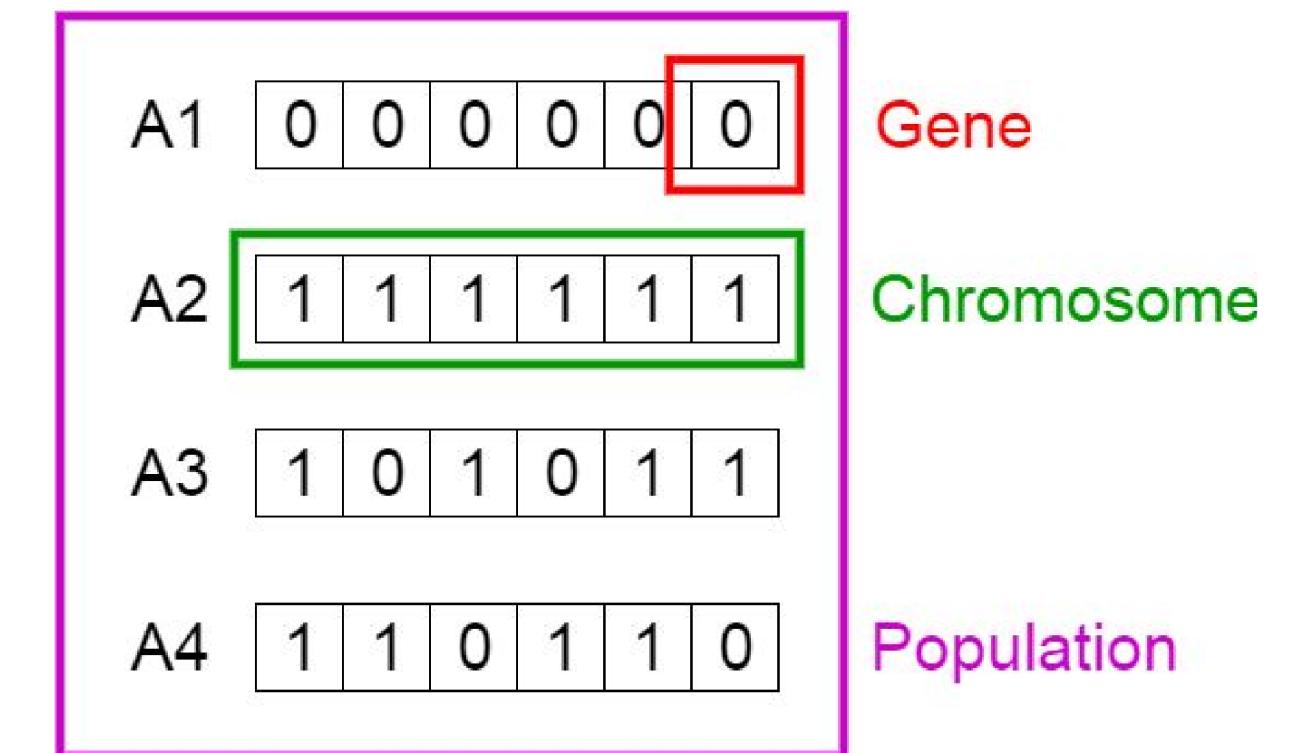
- La fonction de fitness détermine **la qualité d'un individu**, c'est-à-dire sa capacité à rivaliser avec les autres individus.
- Elle attribue un **score** de fitness à chaque individu, évaluant ainsi sa performance par rapport aux critères du problème donné.



Processus d'Algorithmes Génétiques

La sélection:

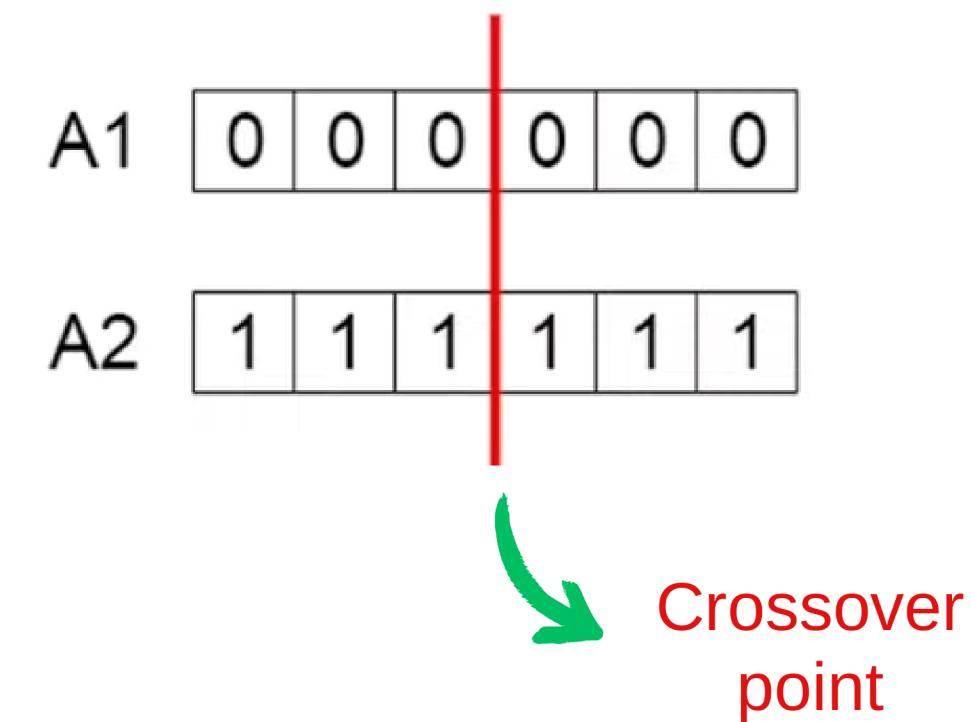
- L'idée de la phase de sélection est de **choisir** les individus les plus aptes et de leur permettre de **transmettre** leurs gènes à la génération suivante.
- Deux paires d'individus (parents) sont sélectionnées en fonction de leurs scores de fitness.
- Les individus ayant une grande aptitude ont plus de chances d'être choisis pour la reproduction.



Processus d'Algorithmes Génétiques

Crossover:

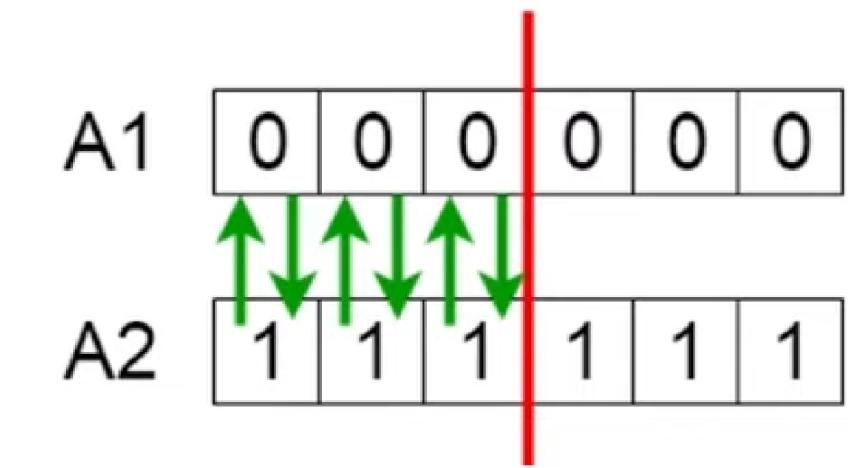
- Le croisement est la phase la plus significative dans un algorithme génétique.
- Pour chaque paire de parents destinés à se reproduire, un **point de croisement** est choisi au hasard à l'intérieur des gènes.
- Par exemple, considérez le point de croisement comme étant 3, comme indiqué.



Processus d'Algorithmes Génétiques

Offspring:

- Les descendants (Offspring) sont créés en échangeant les gènes des parents entre eux jusqu'à ce que le point de croisement soit atteint.
- Les nouveaux descendants sont ajoutés à la population.



A5 1 1 1 0 0 0

A6 0 0 0 1 1 1

Processus d'Algorithmes Génétiques

Mutation:

- Dans certains nouveaux descendants formés, certains de leurs gènes peuvent être soumis à une mutation avec une probabilité aléatoire faible.
- Cela implique que certains des bits dans la chaîne de bits peuvent être inversés.

Before Mutation

A5	1	1	1	0	0	0
----	---	---	---	---	---	---

After Mutation

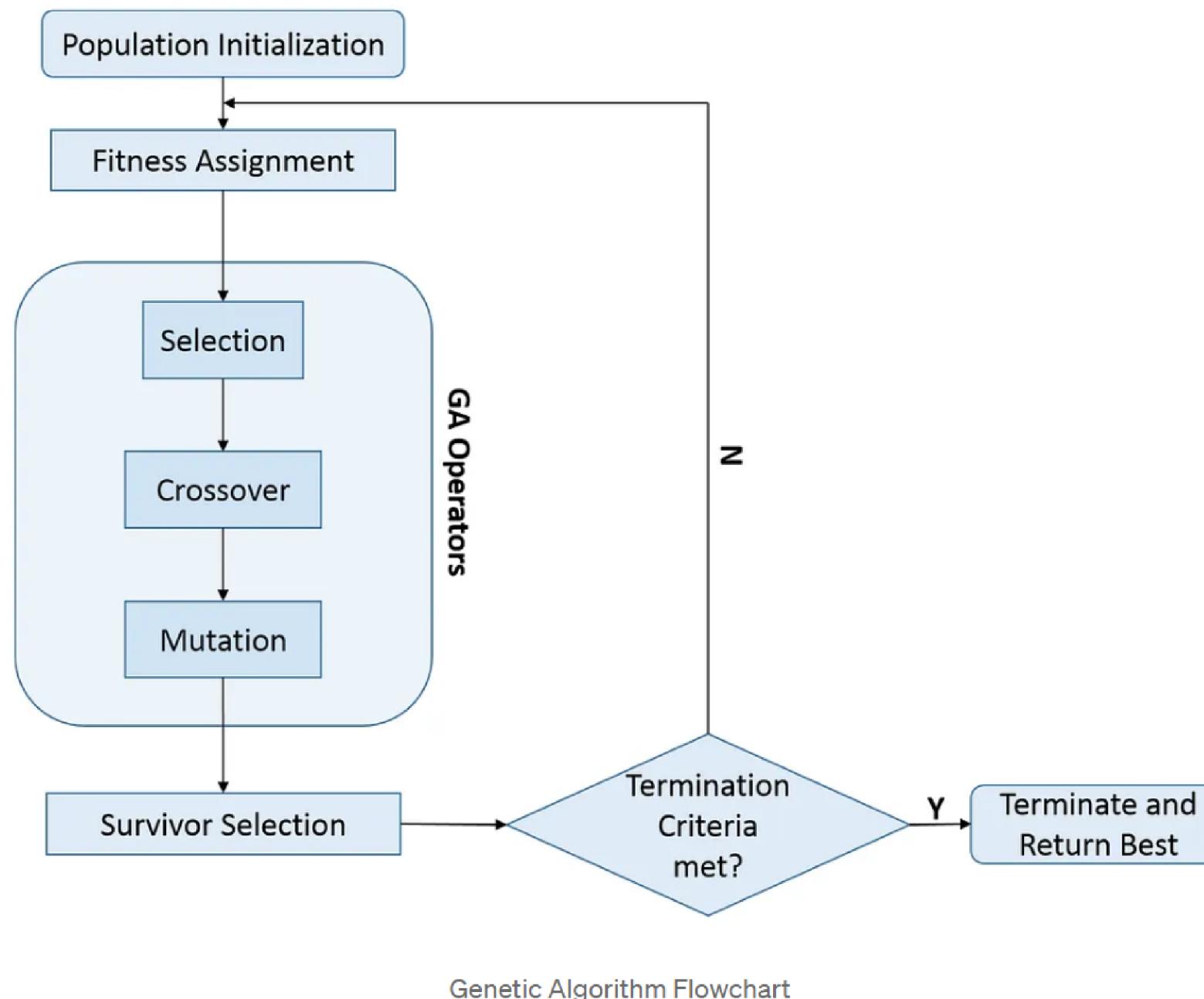
A5	1	1	0	1	1	0
----	---	---	---	---	---	---

Processus d'Algorithmes Génétiques

La terminaison:

- L'algorithme se termine si la population a **convergé** (ne produit pas de descendants qui sont significativement différents de la génération précédente).
- On dit alors que l'algorithme génétique a fourni un ensemble de solutions à notre problème.

Processus d'Algorithmes Génétiques



Genetic Algorithm Flowchart

Processus d'Algorithmes Génétiques

Exemple:

- Maximiser la valeur de la fonction:

$$f(x) = x^2$$

- Les solutions possibles pour x sont limitées à l'intervalle de **0 à 31**.

Processus d'Algorithmes Génétiques

Sélectionner la technique de codage:

- Le minimum est 0 et la valeur maximale est 31. En utilisant un entier binaire de cinq bits, on peut obtenir des nombres entre 0 (00000) et 31 (11111).
- La fonction objective ici est $f(x) = x^2$ qui doit être maximisée.

Processus d'Algorithmes Génétiques

Sélectionner la Population Initiale:

- Pour commencer, sélectionnez la population initiale de manière aléatoire.
- Ici, une population initiale de taille 4 a été choisie, mais n'importe quel nombre de populations peut être sélectionné en fonction des besoins et de l'application.
- Le tableau montre une population initiale sélectionnée de manière aléatoire.

Processus d'Algorithmes Génétiques

Calculer la fonction de fitness

String No.	Initial Population	X value	Fitness $f(x) = x^2$	Prob	%Prob	Expected count	Actual count
1	01100	12					
2	11001	25					
3	00101	5					
4	10011	19					
Sum							
Average							
Maximum							

Processus d'Algorithmes Génétiques

Calculer la fonction de fitness

String No.	Initial Population	X value	Fitness $f(x) = x^2$	Prob	%Prob	Expected count	Actual count
1	01100	12	144				
2	11001	25	625				
3	00101	5	25				
4	10011	19	361				
Sum			1155				
Average			288.75				
Maximum			625				

Processus d'Algorithmes Génétiques

Calculer la fonction de fitness

String No.	Initial Population	X value	Fitness $f(x) = x^2$	Prob	%Prob	Expected count	Actual count
1	01100	12	144	0.1247			
2	11001	25	625	0.5411			
3	00101	5	25	0.0216			
4	10011	19	361	0.3126			
Sum			1155	1.0			
Average			288.75	0.25			
Maximum			625	0.5411			

Processus d'Algorithmes Génétiques

Calculer la fonction de fitness

String No.	Initial Population	X value	Fitness $f(x) = x^2$	Prob	%Prob	Expected count	Actual count
1	01100	12	144	0.1247	12.47		
2	11001	25	625	0.5411	54.11		
3	00101	5	25	0.0216	2.16		
4	10011	19	361	0.3126	31.26		
Sum			1155	1.0	100		
Average			288.75	0.25	25		
Maximum			625	0.5411	54.11		

Processus d'Algorithmes Génétiques

$$\text{Expected Count} = \frac{f(x_i)}{\text{Avg}(\sum f(x))}$$

Processus d'Algorithmes Génétiques

Calculer la fonction de fitness

String No.	Initial Population	X value	Fitness $f(x) = x^2$	Prob	%Prob	Expected count	Actual count
1	01100	12	144	0.1247	12.47	0.4987	1
2	11001	25	625	0.5411	54.11	2.1645	2
3	00101	5	25	0.0216	2.16	0.0866	0
4	10011	19	361	0.3126	31.26	1.2502	1
Sum			1155	1.0	100	4	4
Average			288.75	0.25	25	1	1
Maximum			625	0.5411	54.11	2.1645	2

Processus d'Algorithmes Génétiques

String No.	Mating Pool	Crossover Point	Offspring after crossover	X value	Fitness $f(x) = x^2$
1	01100	4	01101	13	169
2	11001		11000	24	576
3	11001	2	11011	27	729
4	10011		10001	17	289
Sum					1763
Average					440.75
Maximum					729

Processus d'Algorithmes Génétiques

String No.	Offspring after crossover	Mutation Chromosome for flipping	Offspring after mutation	X value	Fitness $f(x) = x^2$
1	01101	10000	11101	29	841
2	11000	00000	11000	24	576
3	11011	00000	11011	27	729
4	10001	00101	10100	20	400
Sum					2546
Average					636.5
Maximum					841

05

Intensification et Diversification



Intensification et Diversification

- l'intensification et la diversification sont deux concepts clés dans les algorithmes génétiques qui peuvent avoir un impact significatif sur le succès de l'algorithme.
- L'intensification implique de se concentrer sur l'exploitation des meilleures solutions actuelles, tandis que la diversification implique d'explorer de nouvelles zones de l'espace de recherche. Les deux sont importants pour obtenir des résultats optimaux et éviter une convergence prématuée.
- En équilibrant ces deux stratégies, les algorithmes génétiques peuvent résoudre efficacement des problèmes d'optimisation complexes.

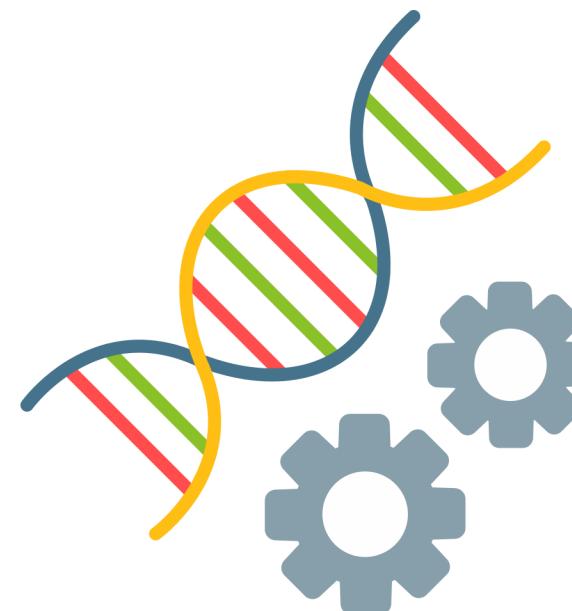


Intensification

Les opérations d'intensification se concentrent sur l'exploitation des solutions existantes. Les paramètres clés responsables de l'intensification comprennent :

Mécanisme de Sélection

Peuvent être utilisés pour intensifier la recherche en choisissant les meilleures solutions de la population et en propageant leurs caractéristiques à la génération suivante



Les Opérateurs de Croisement

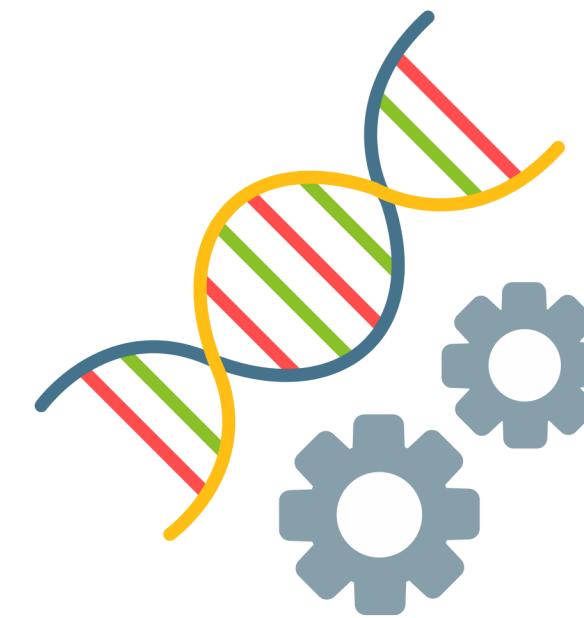
Ils jouent un rôle crucial dans l'exploitation des caractéristiques prometteuses des solutions existantes, contribuant ainsi de manière significative à l'intensification du processus évolutif

Diversification

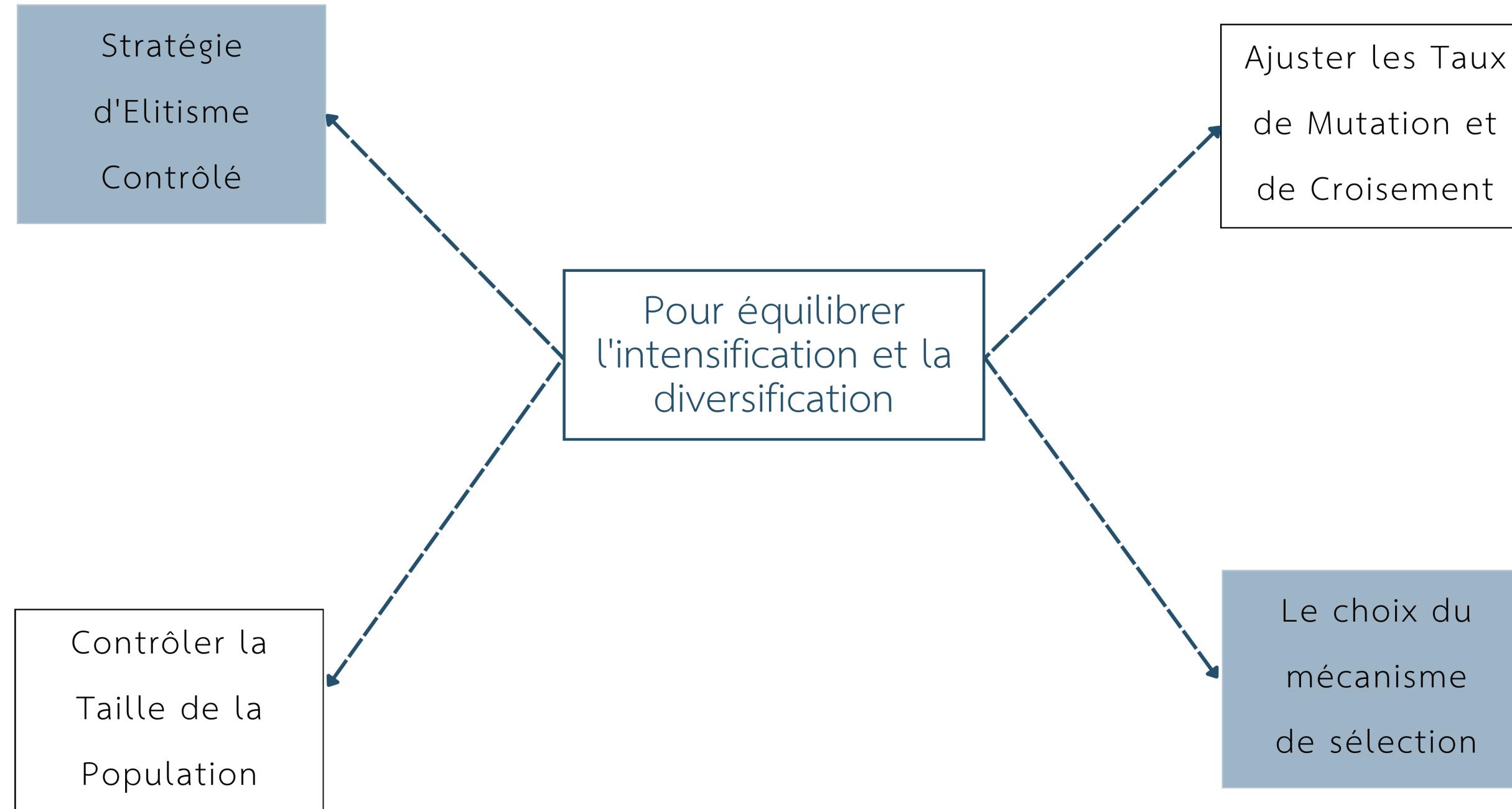
Les opérations de diversification explorent de nouvelles solutions. Les paramètres qui contribuent à la diversification sont :

Les Opérateurs de Mutation

la mutation perturbe les caractéristiques génétiques des individus de la population, créant ainsi une diversité qui peut être cruciale pour l'exploration de solutions.



Intensification et Diversification



06

Applications



Applications

Problème du Voyageur de Commerce (TSP)

Les algorithmes génétiques sont appliqués pour résoudre le problème du voyageur de commerce, qui est un problème bien connu en optimisation combinatoire.

JSP (Job Shop Scheduling Problem)

Les algorithmes génétiques peuvent être utilisés pour optimiser l'ordonnancement des tâches dans un atelier de production.

Sac à dos (Knapsack problem)

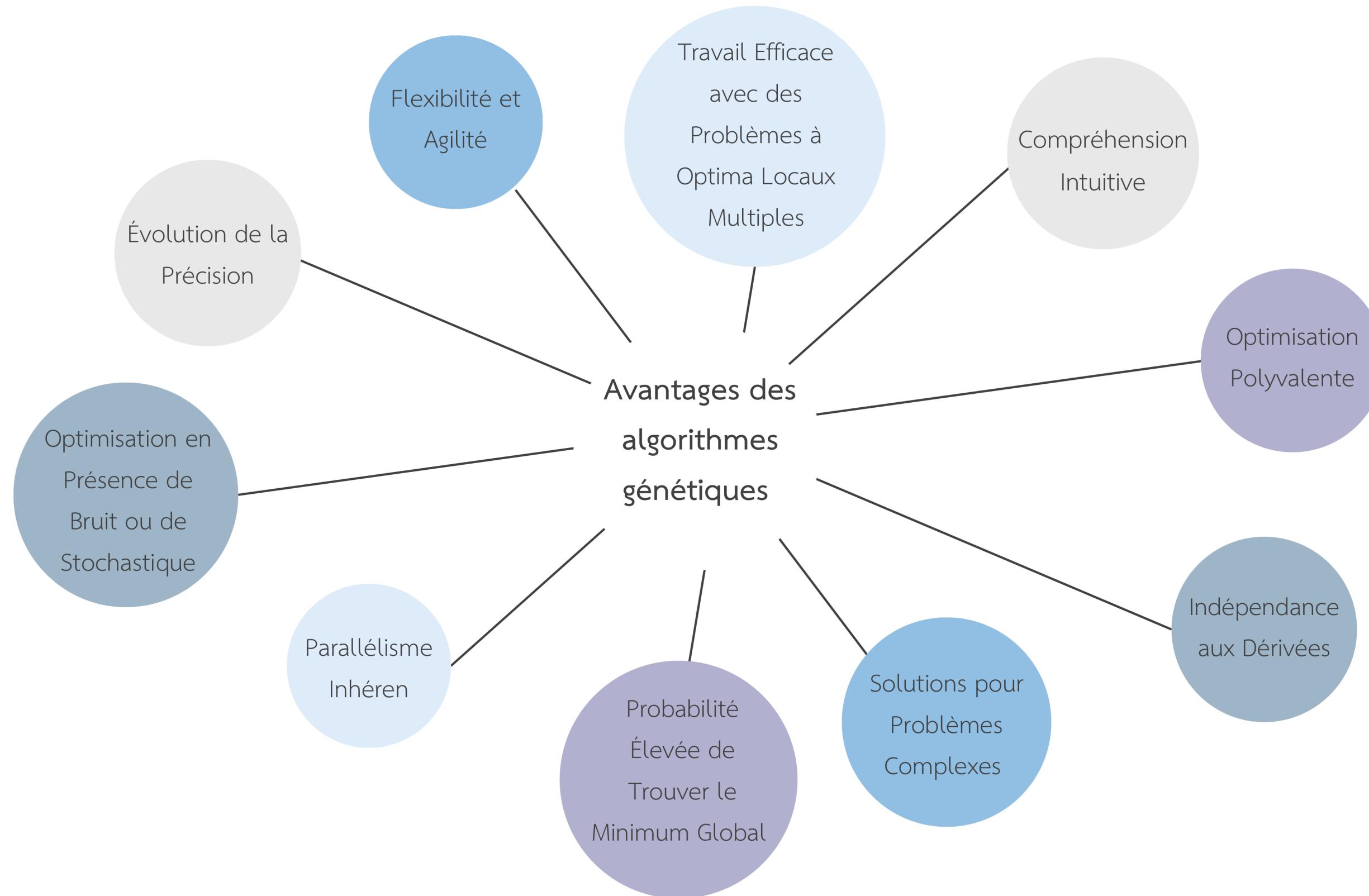
L'utilisation d'algorithmes génétiques pour le problème du sac à dos permet d'explorer efficacement l'espace des solutions, en particulier lorsque le nombre d'objets est important.

07

Avantages et limitations



Avantages



Limitations

Les algorithmes génétiques (AG) présentent un certain nombre de limites telles que :

Coût Computationnel Élevé

Car il nécessite l'exploration de l'espace de recherche en examinant de multiples permutations et combinaisons avant d'atteindre le résultat souhaité.

Nature Stochastique et Temps de Convergence

En raison de sa nature stochastique et de la nécessité d'effectuer de multiples itérations pour trouver la solution.

Efficacité Diminuée pour les Problèmes Simples

Bien qu'efficace pour résoudre des problèmes complexes, l'algorithme est contre-productif et inefficace pour des problèmes simples.

Limitations

Les algorithmes génétiques (AG) présentent un certain nombre de limites telles que :

Complexité d'Implémentation

Bien que le concept soit facile à comprendre, l'implémentation de l'algorithme génétique peut être difficile et parfois plus artistique.

Difficultés dans la Conception

Bien que moins d'informations de l'utilisateur soient nécessaires sur le problème, cet avantage est compensé par les difficultés.

Absence de Garantie sur la Qualité des Résultats

Il n'y a aucune garantie de la qualité des résultats générés par de tels algorithmes.

08

Conclusion



Conclusion

- les algorithmes génétiques représentent une approche puissante et polyvalente pour résoudre une variété de problèmes d'optimisation. Inspirés par les mécanismes de l'évolution naturelle, ces algorithmes combinent la sélection, le croisement et la mutation pour guider la recherche vers des solutions potentiellement optimales.
- L'équilibre délicat entre intensification et diversification est essentiel pour garantir le succès de l'approche.
- En exploitant ces principes, les algorithmes génétiques ont démontré leur efficacité dans des domaines variés tels que l'ingénierie, l'optimisation des processus, la planification, et bien d'autres.



09

Références



Références

- Applying the genetic algorithm to optimization problems H. Wan Department of Computer Studies, Lingnan College, Hong Kong
- Holland, J.H. Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1975.
- https://sci-hub.se/10.1007/978-3-642-53856-8_46
- <https://ijcsit.com/docs/Volume%205/vol5issue03/ijcsit20140503404.pdf>
- <https://www.geeksforgeeks.org/crossover-in-genetic-algorithm/>
- <https://www.ijert.org/research/parent-selection-operators-for-genetic-algorithms-IJERTV2IS110523.pdf>
- https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_application_areas.htm
- <https://www.whitman.edu/documents/academics/mathematics/2014/carrjk.pdf>
- <https://www.analytixlabs.co.in/blog/genetic-algorithm/>

10

Implémentation

