



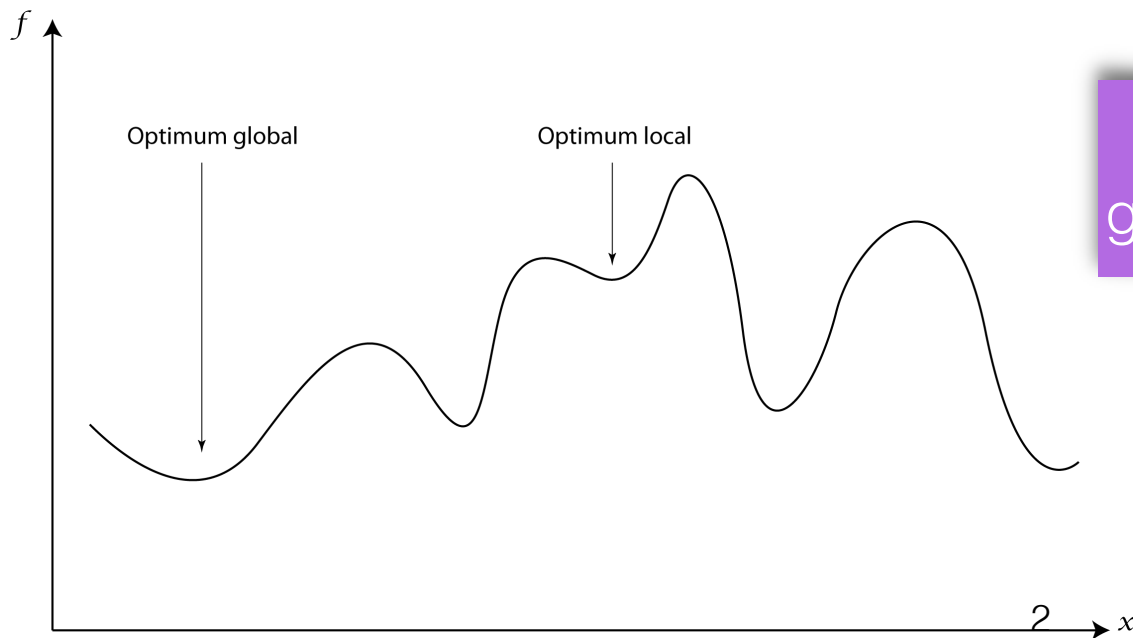
CentraleSupélec

# ST2 — Théorie des Jeux

## **EI — Jeux Évolutionnaires**

# Optimisation

- Optimiser (minimiser/maximiser) une fonction
  - Parcours, emplois du temps, traitement d'image...
- Explorer un espace de recherche



petit → méthode déterministe  
grand → méthode stochastique

# Optimisation

- « Essai et erreur »
  - Solutions admissibles
  - Solutions coût optimal local

- Analyser le problème

- variables

**petit** si le système est simple  
si le concepteur sait quelles variables  
faire évoluer

**grand** si le système est complexe  
si le concepteur ne sait pas quelles variables  
faire évoluer

- espace de recherche

**infini** → **fini**

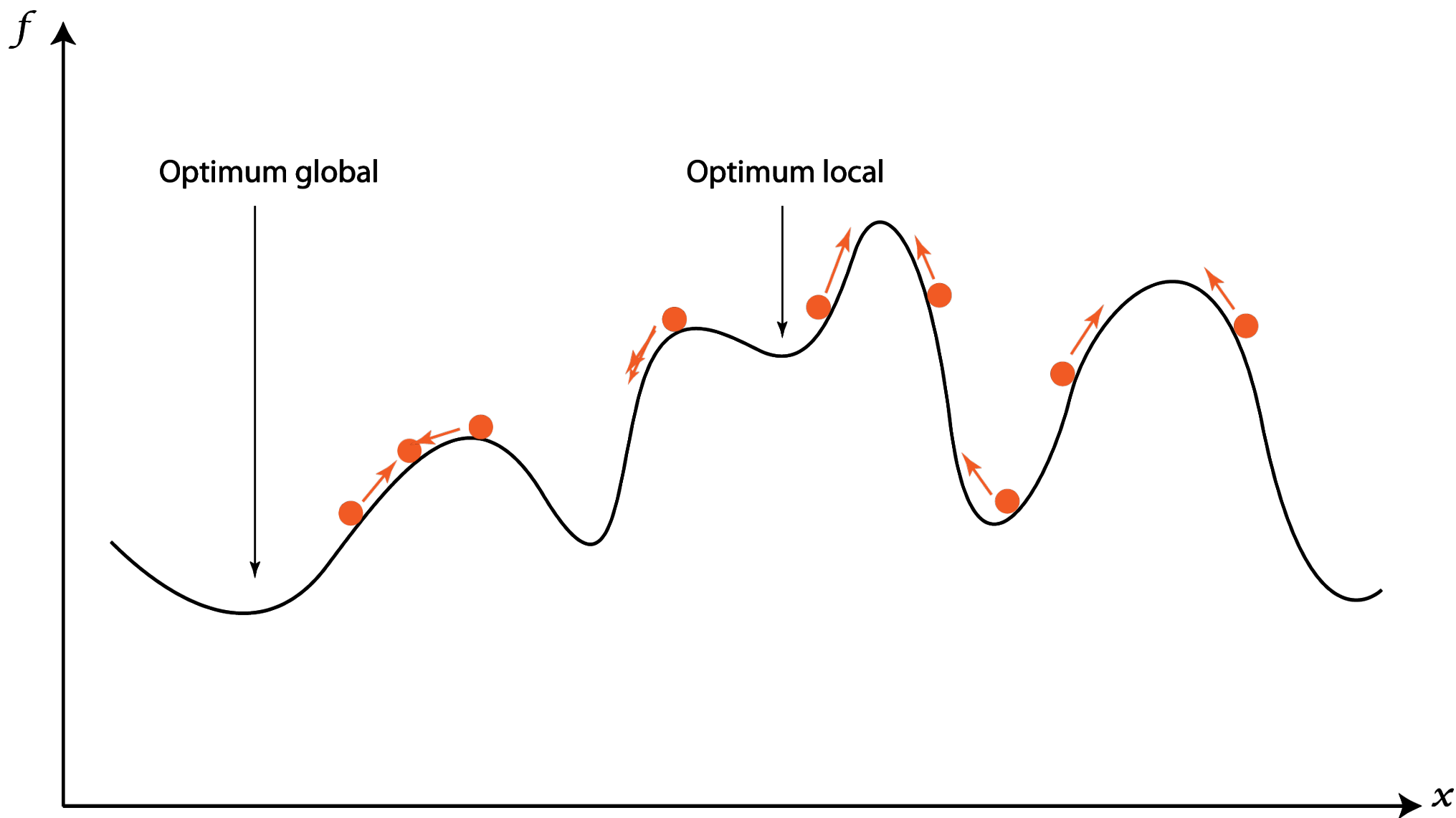
- objectif

**fitness** → **mesure**

- méthode d'optimisation

**déterministe**

**non-déterministe**



stochastique

non-déterministe

répétition

simple

# Monte-Carlo

## Estimation de la surface d'un lac

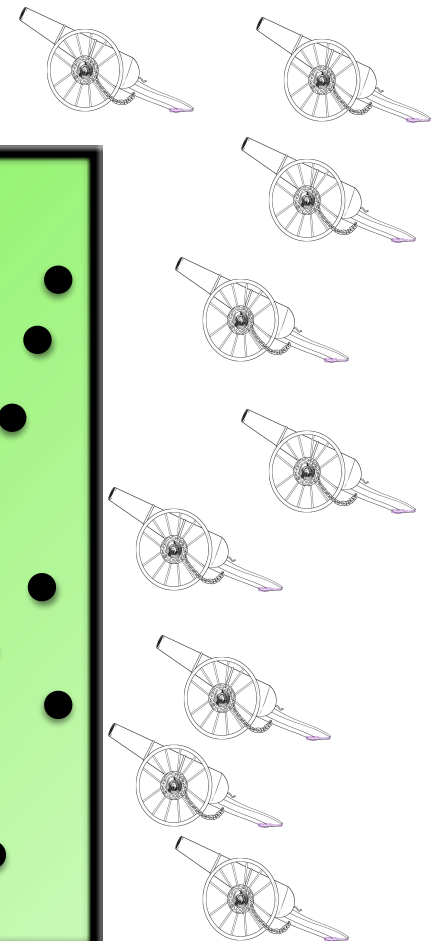
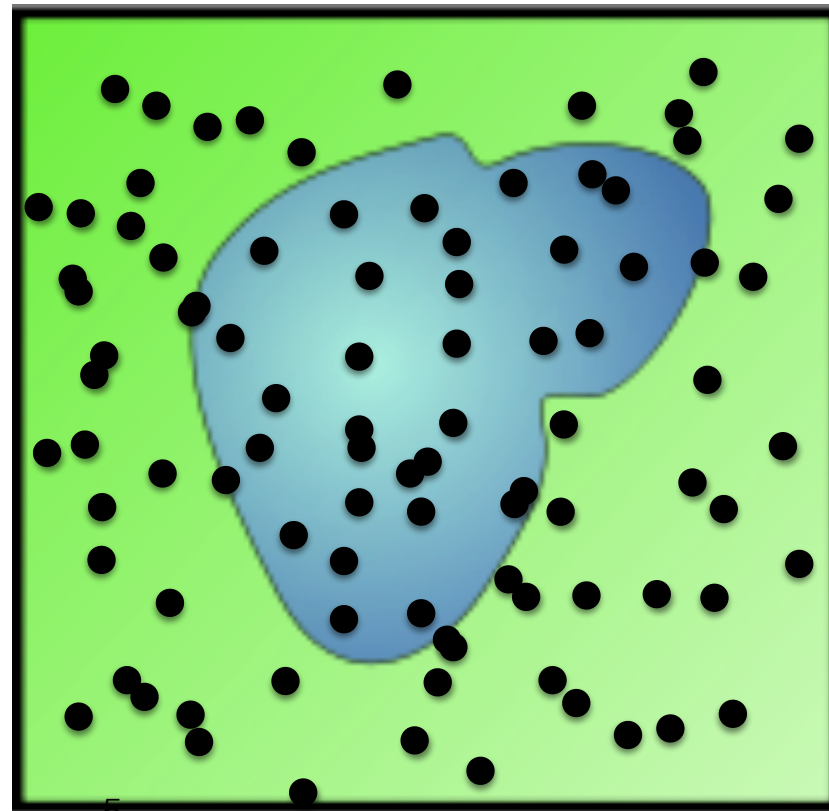
1. Tracer un terrain dont on connaît la superficie autour du lac
2.  $X$  tirs au hasard
3.  $N$  boulets sur le terrain
4.  $X-N$  boulets dans le lac

$$\frac{SUPERFICIE_{terrain}}{SUPERFICIE_{lac}} = \frac{X}{X - N}$$

$$\Rightarrow SUPERFICIE_{lac} = \frac{X - N}{X} \times SUPERFICIE_{terrain}$$

**Volume des données**

**Qualité du tirage aléatoire**



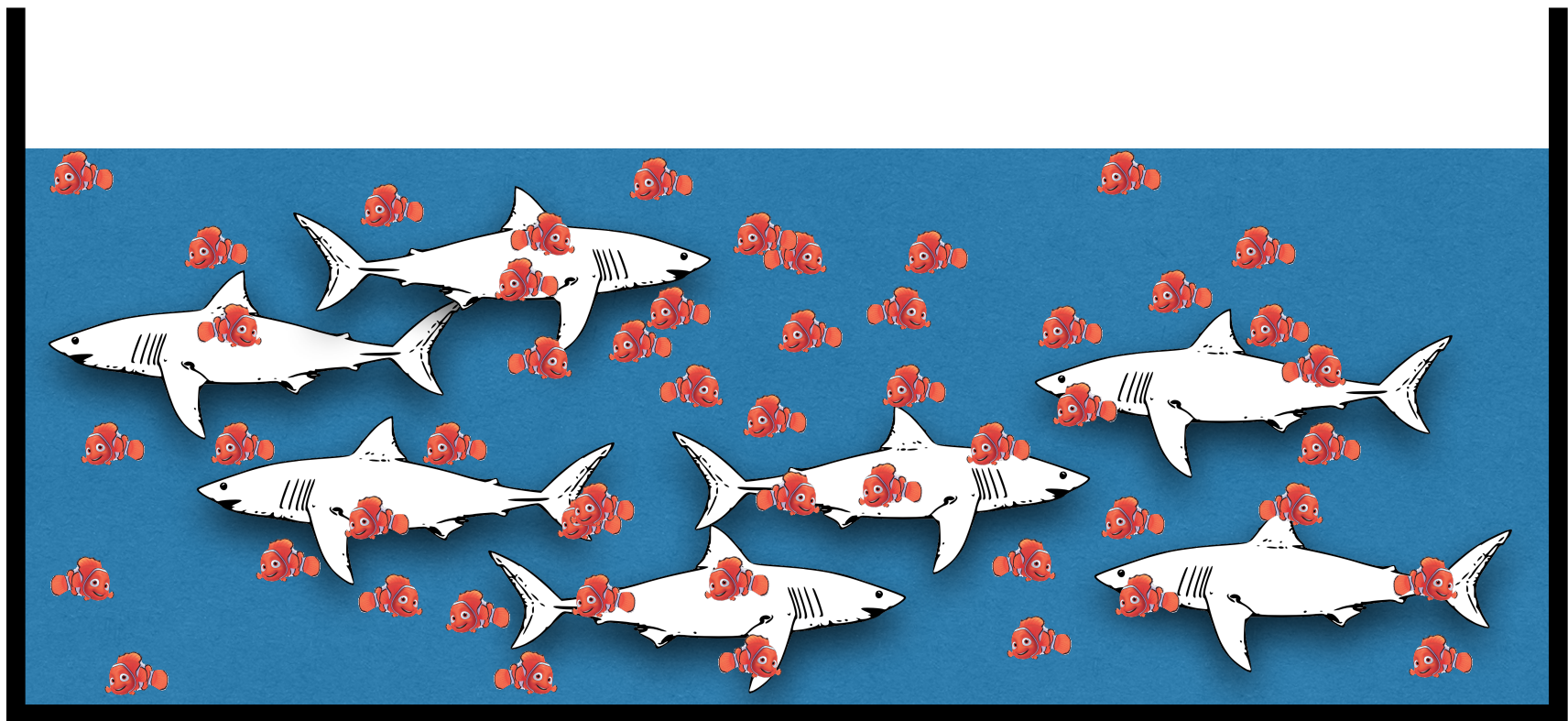
# Biologie

- Étude du vivant
- Spectre très large
  - Formes très variées, nombreuses échelles

*En quoi l'étude du vivant peut-elle nous aider à résoudre  
un problème d'optimisation ?*

- Le vivant s'adapte constamment à son environnement

*Optimisation de la fonction « survie »*



# Algorithme génétique

## **CONSTAT 1**

espace de recherche grand = temps de calcul grand

## **CONSTAT 2**


vivant optimise la fonction « survie »

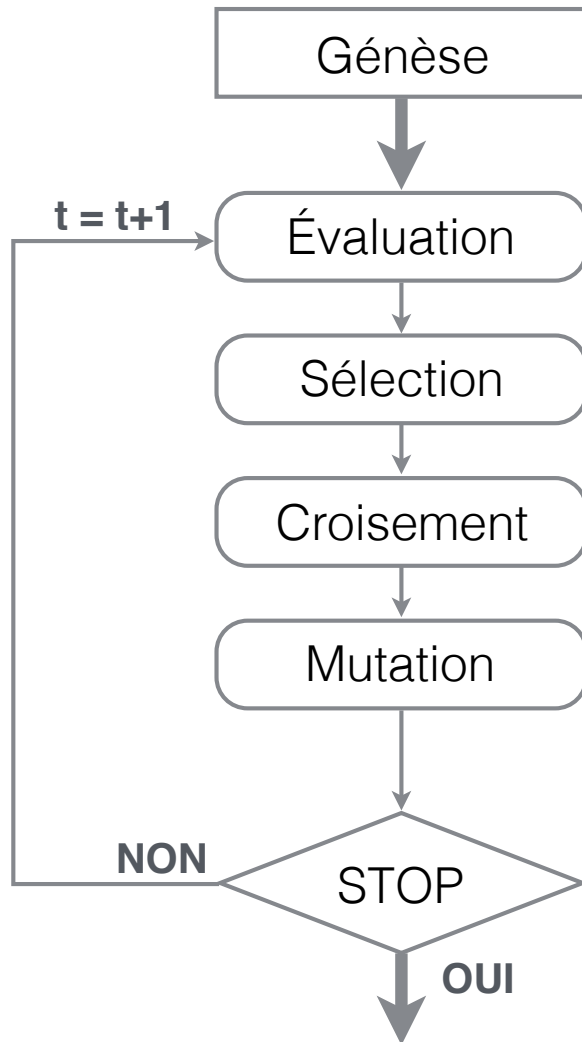
## **PROPOSITION**

s'inspirer du vivant

pour résoudre des problèmes d'optimisation

# Algorithme génétique

- Individu = point dans l'espace de recherche
- Valeur du critère à optimiser (adaptation)
- Population
  - Sélection ——— meilleurs éléments
  - Croisement ——— exploitation des résultats déjà obtenus
  - Mutation ——— exploration de l'espace de recherche
- Arrêt 
  - nombre d'itérations
  - convergence



Aucune connaissance

Solution proche de l'optimal

Solutions meilleures  
que par approche classique

# Le voyageur de commerce

## Wikipédia

*En informatique, le problème du voyageur de commerce est un problème d'optimisation qui, étant donnée une liste de villes, et des distances entre toutes les paires de villes, détermine un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ.*

NP-complet

# Le voyageur de commerce

**Énumération**

**$O(n!)$**

10 villes → 0,18 s

15 villes → 12 h

20 villes → 2 000 ans

**Programmation Dynamique**

**$O(n^2 \cdot 2^n)$**

10 villes → 0,1 s

30 villes → 11 j

40 villes → 55 ans

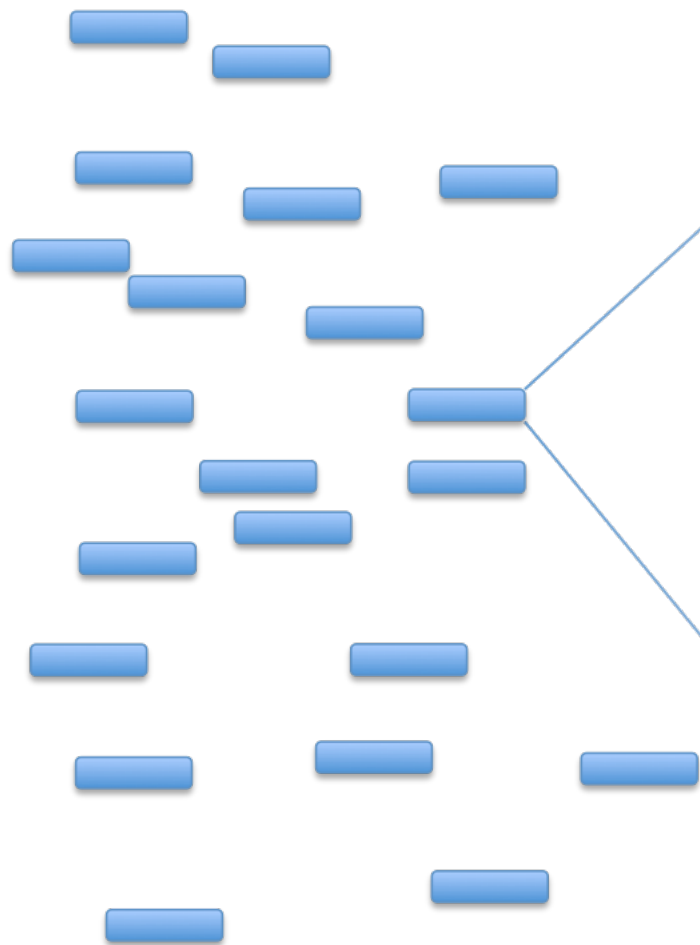
**AG**    **n** · **i** ind. · **g** gén.

**$O(n)$**

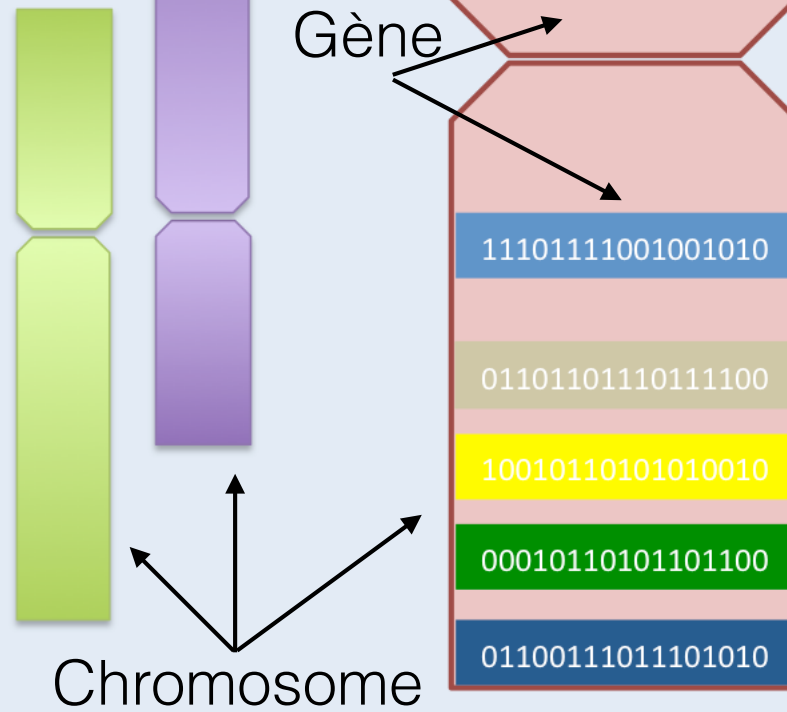
10 villes → 1s

100 villes → 10s

# Population



## Individu



# Encodage des gènes

Réels, entiers, binaire, Gra

Binaire



BIT LE PLUS SIGNIFIANT

$$g_i = \sum_{j=0}^{31} b_j \cdot 2^j \quad 0 \leq g_i \leq g_{max}, \forall i \in [1, n]$$

encodage / décodage

$$g_i = \frac{x_i - x_{imin}}{x_{imax} - x_{imin}} \cdot g_{imax}$$

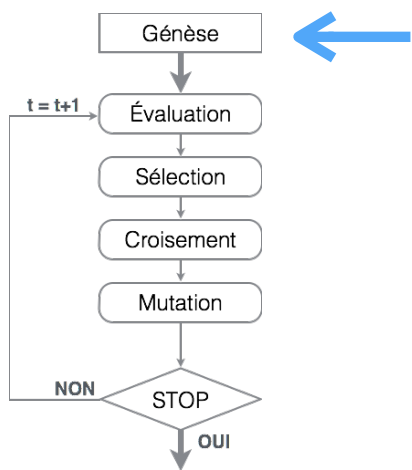
$$x_i = x_{imin} + (x_{imax} - x_{imin}) \cdot \frac{g_i}{g_{max}}$$

Application au voyageur de commerce

Un **gène** représente une ville et est codé par un entier.

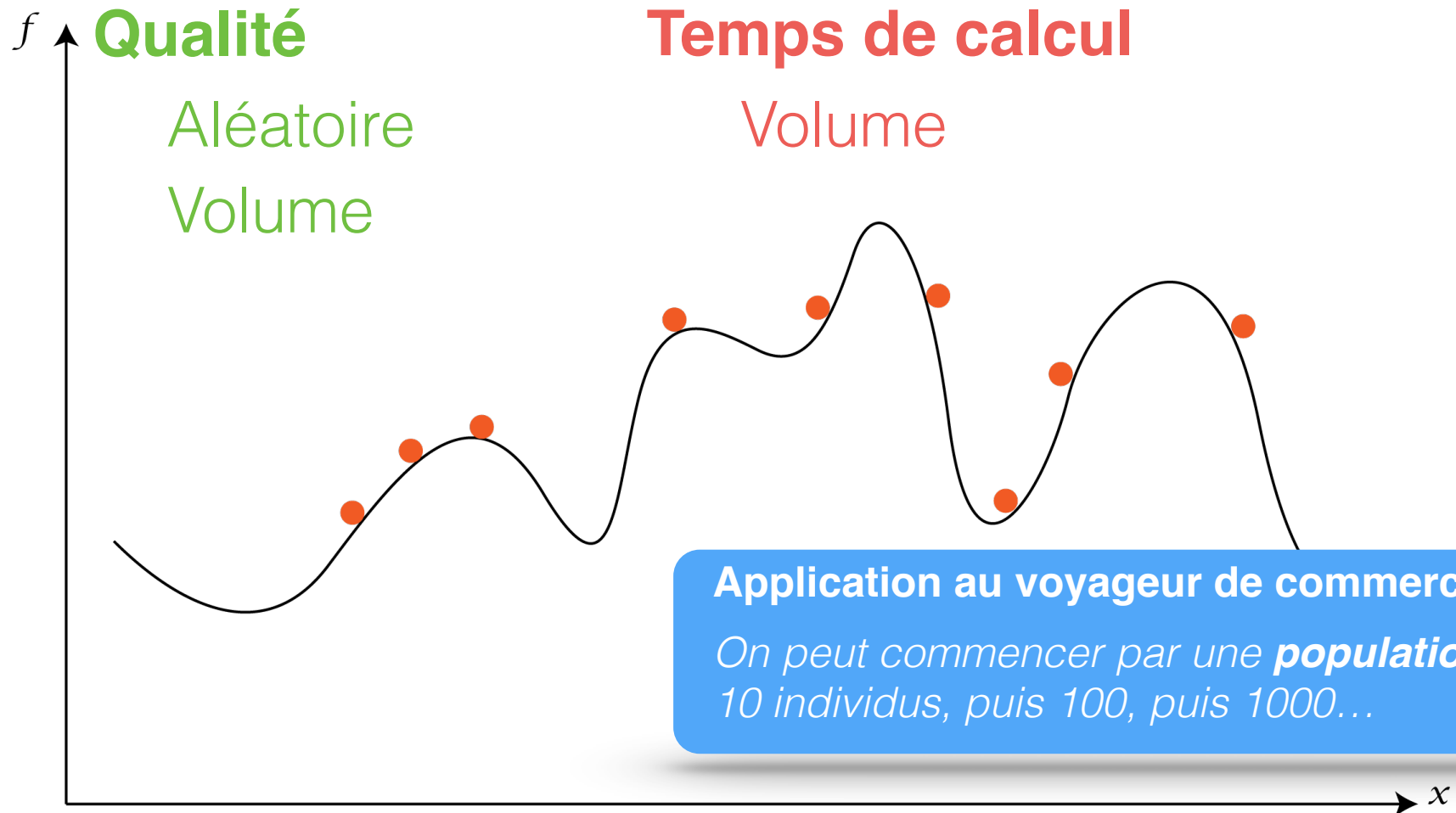
Application au voyageur de commerce

Un **chromosome** représente un chemin parmi les villes du problème.

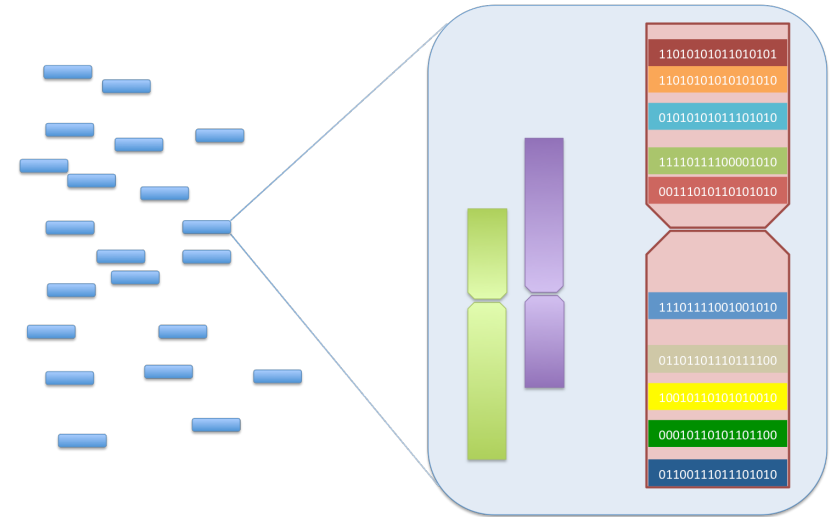
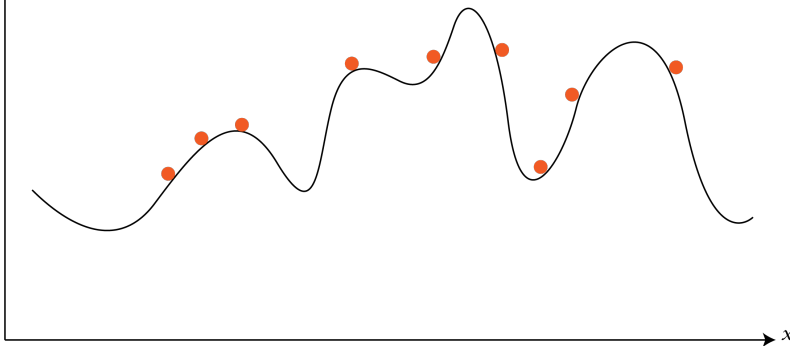
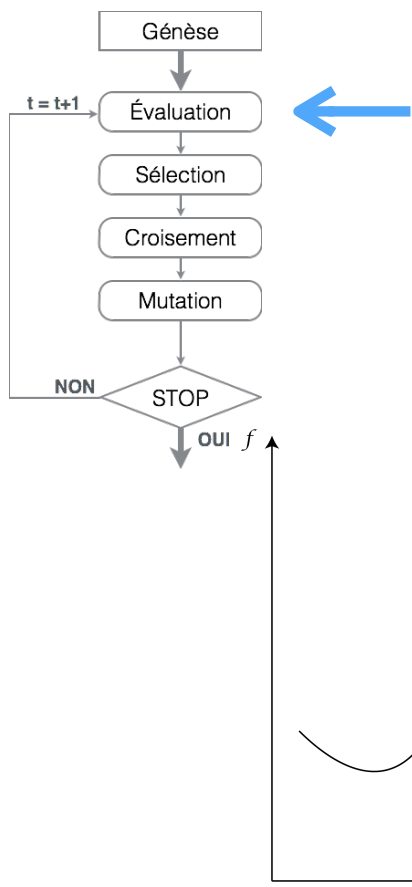


# ÉTAPE 1

## GÉNÉRATION INITIALE



# ÉTAPE 2 ÉVALUATION



## Application au voyageur de commerce

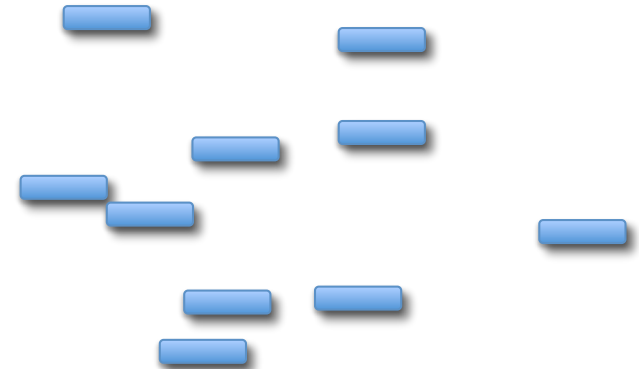
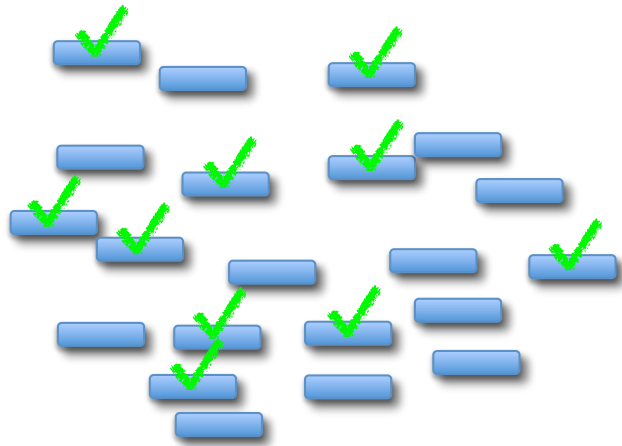
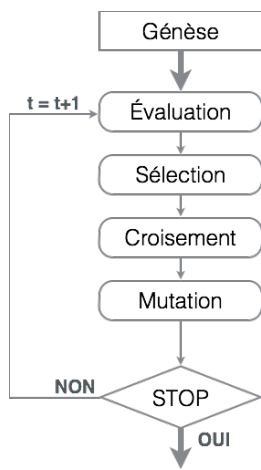
La fonction de **fitness** calculera la distance parcourue si le voyageur emprunte le chemin représenté par l'individu évalué.

INDIVIDU



SCORE

# ÉTAPE 3 SÉLECTION



# ÉTAPE 3

## SÉLECTION PAR ÉLITISME

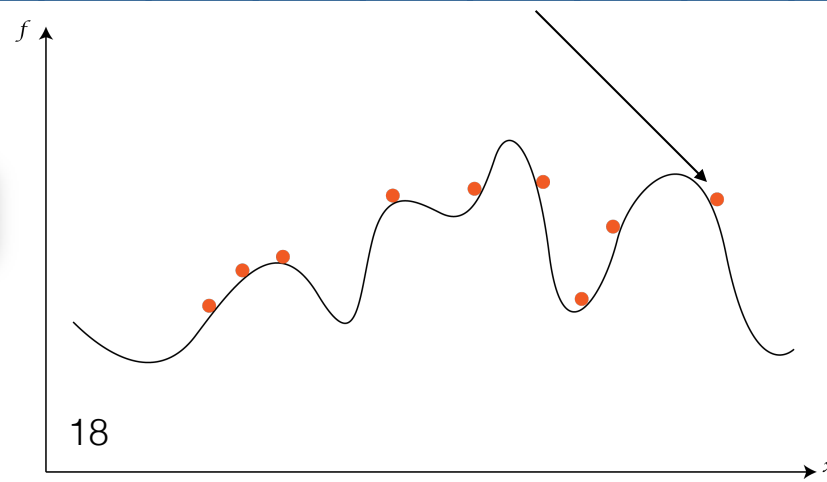


**Sélectionnés**

**Rejetés**

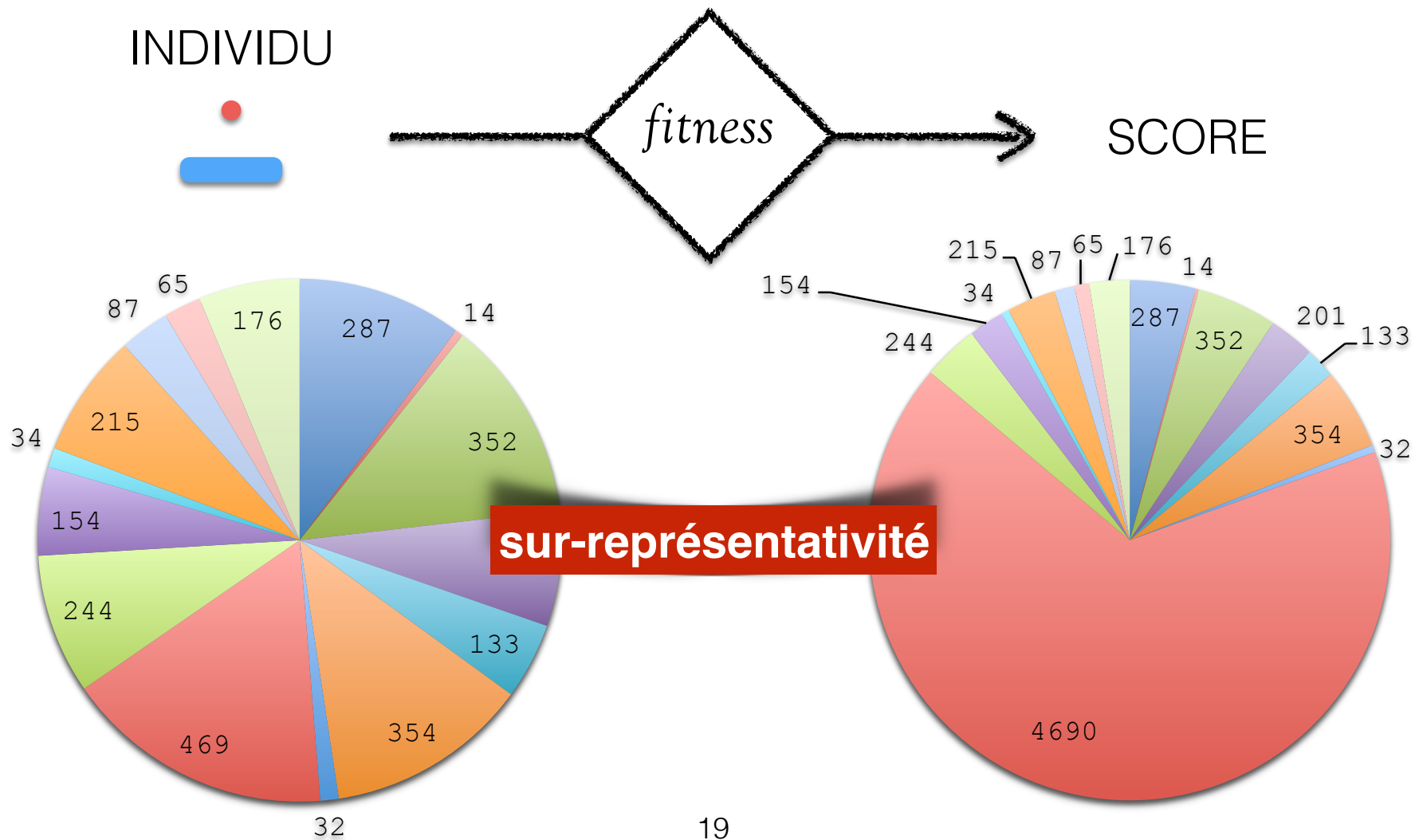
469	354	352	287	244	215	201	176	154	133	87	65	34	32	14
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	----	----	----	----	----

**Forte pression de sélection**



# ÉTAPE 3

## SÉLECTION PAR ROULETTE



# ÉTAPE 3

## SÉLECTION PAR RANG

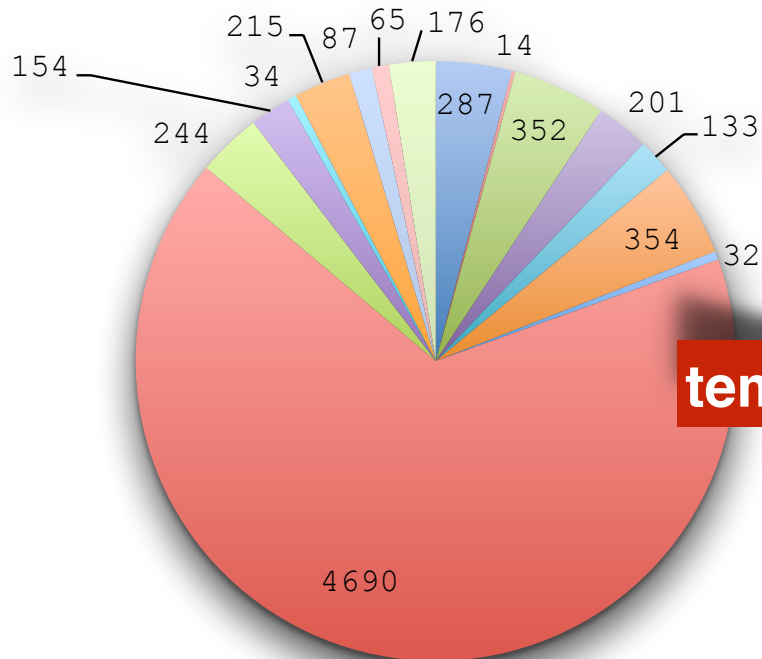
INDIVIDU



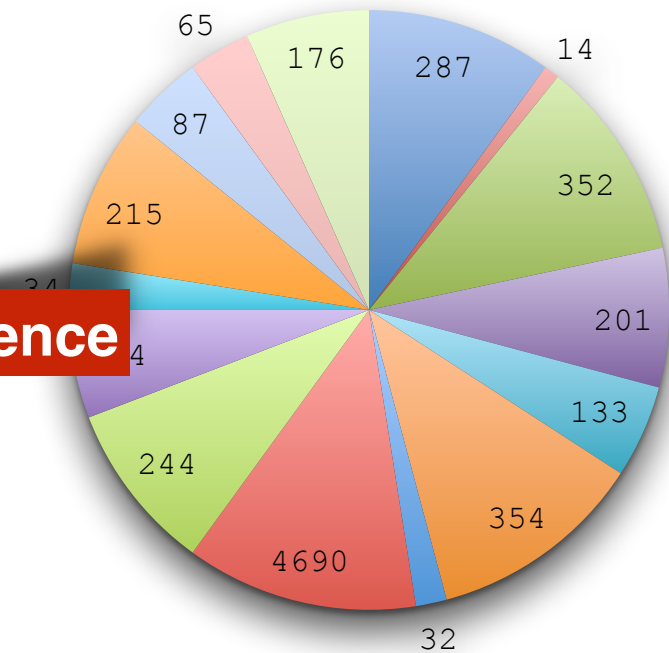
*fitness*

SCORE

6	10	15	3	7	12	1	8	13	9	5	14	11	2	4
133	215	4690	34	154	287	14	176	352	201	87	354	244	32	65



**temps de convergence**



# ÉTAPE 3

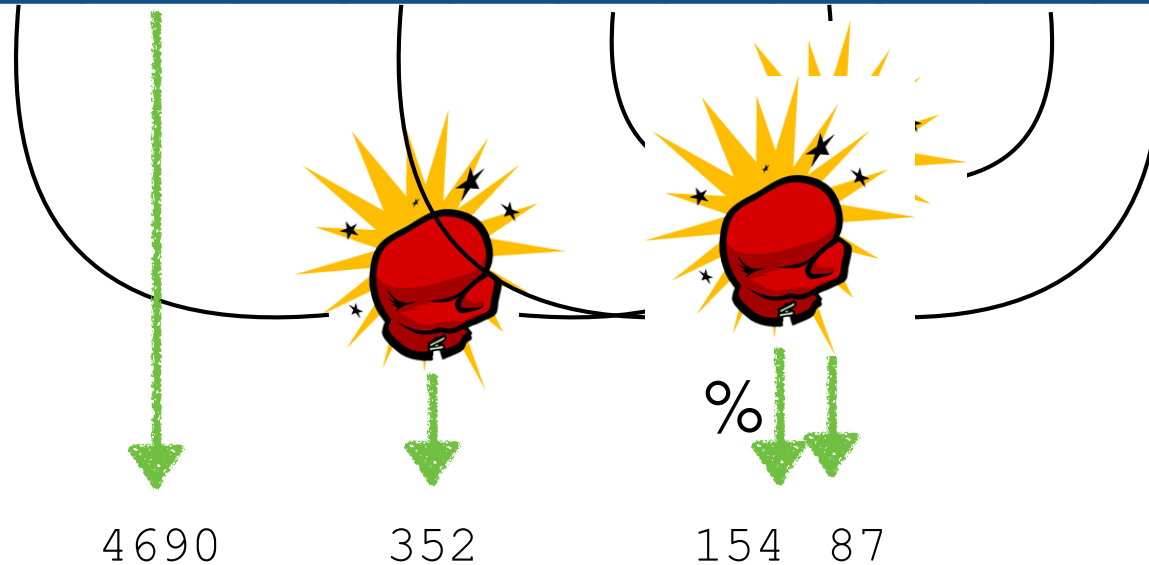
## SÉLECTION PAR TOURNOI

INDIVIDU

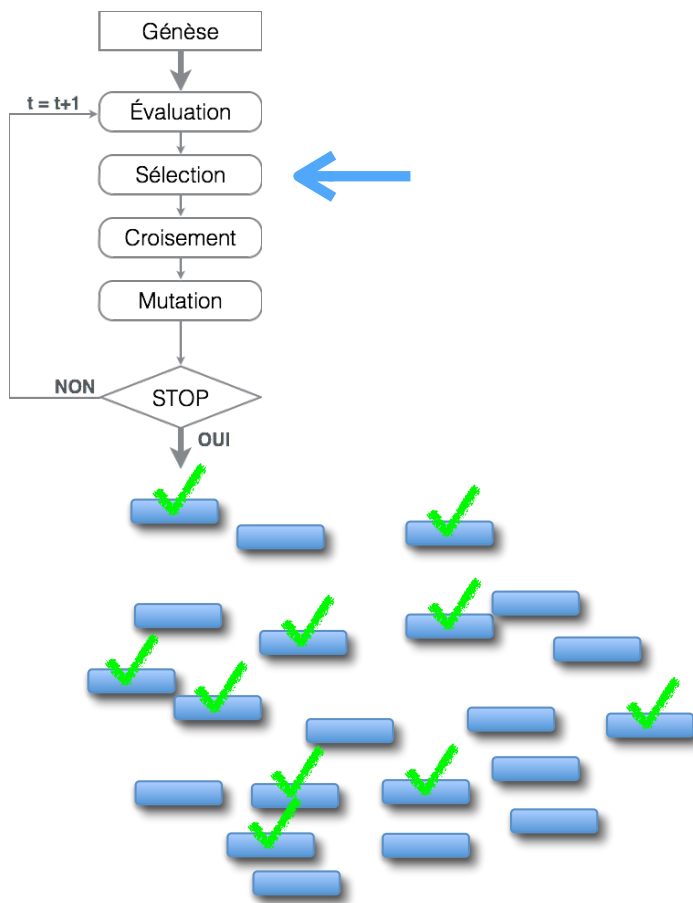


SCORE

133	215	4690	34	154	287	14	176	352	201	87	354	244	32	65
-----	-----	------	----	-----	-----	----	-----	-----	-----	----	-----	-----	----	----



# ÉTAPE 3 SÉLECTION

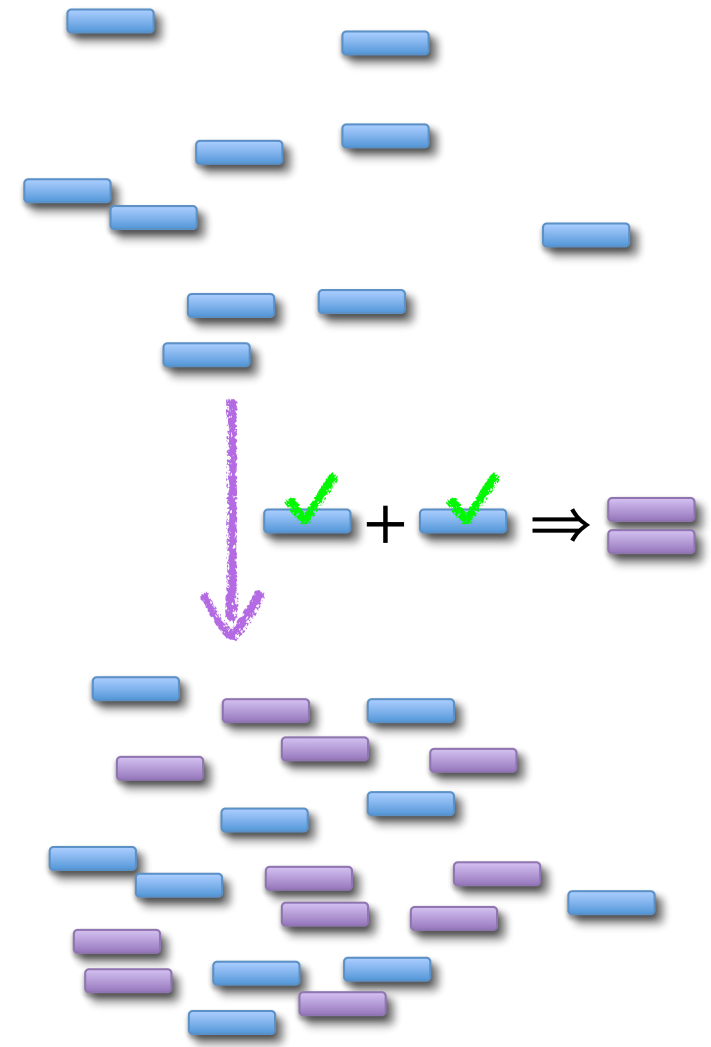
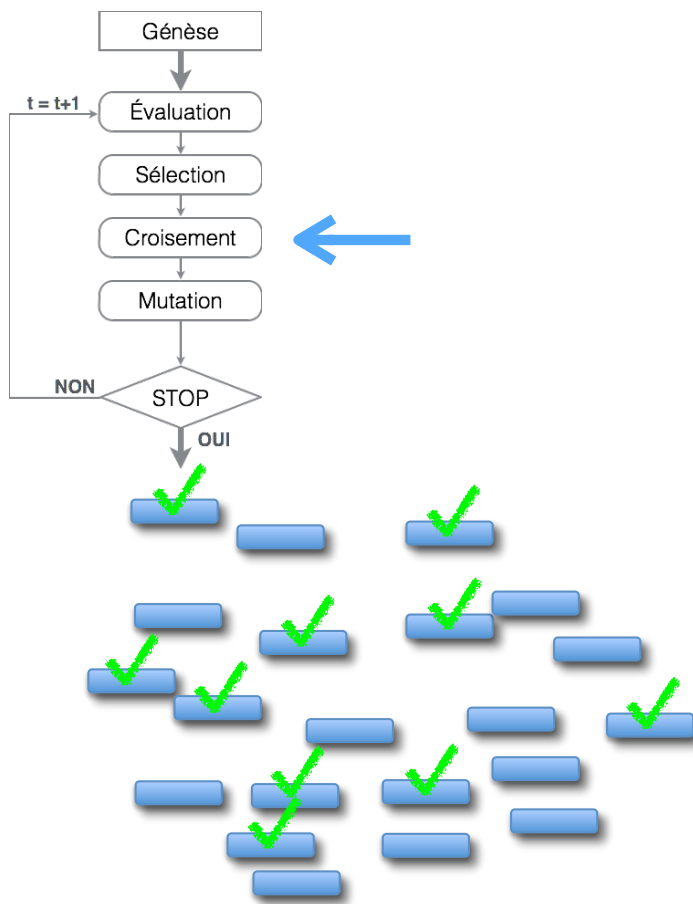


## Application au voyageur de commerce

La méthode de **sélection** est au choix du concepteur.

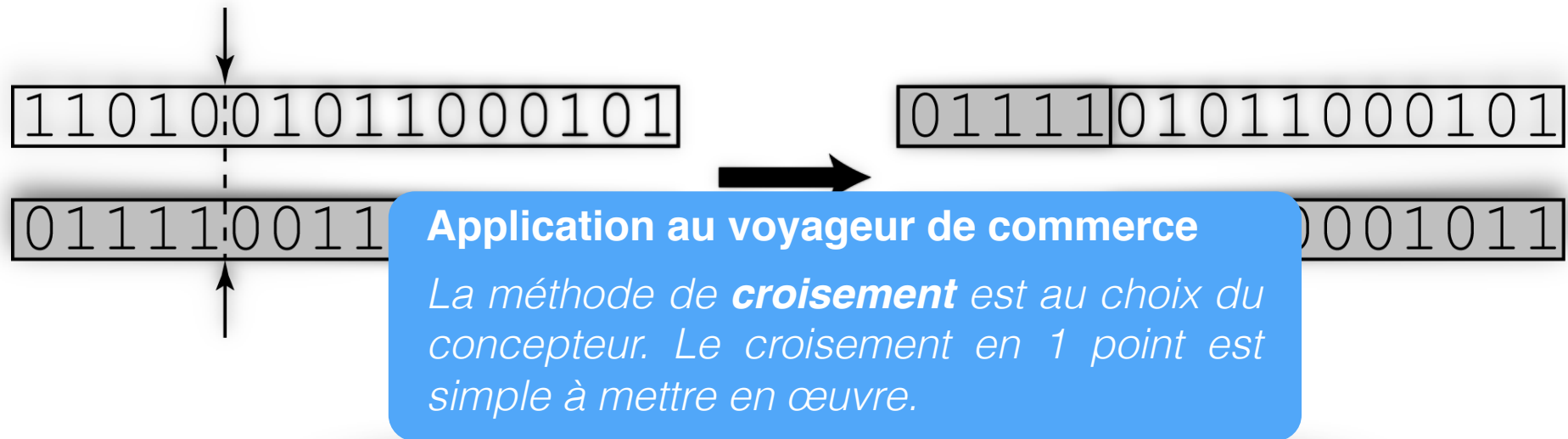
# ÉTAPE 3

## CROISEMENT

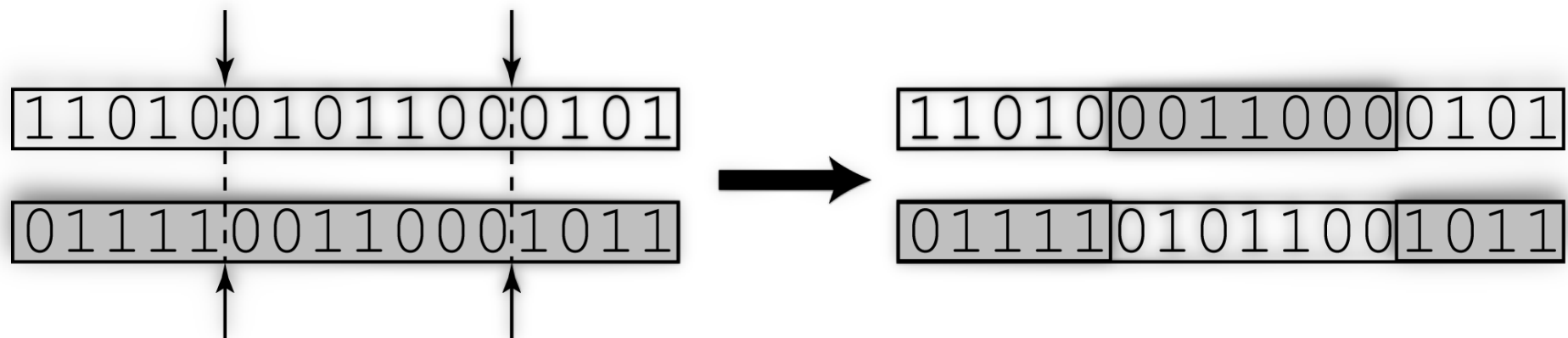


# ÉTAPE 4

## CROISEMENT EN 1 POINT

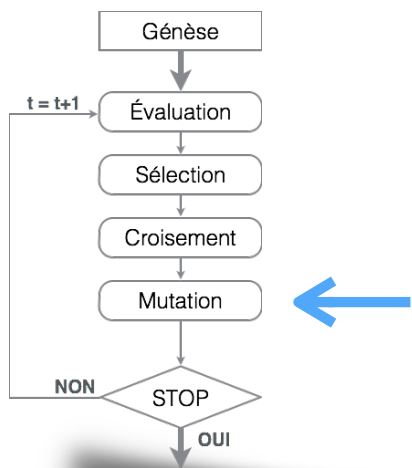


## CROISEMENT EN 2 POINTS



# ÉTAPE 5

## MUTATION



**Stagnation de l'évolution**

**Dérive génétique**

1101001011000101



1101001011010101

$$\Rightarrow 0,001 < p_m < 0,01$$

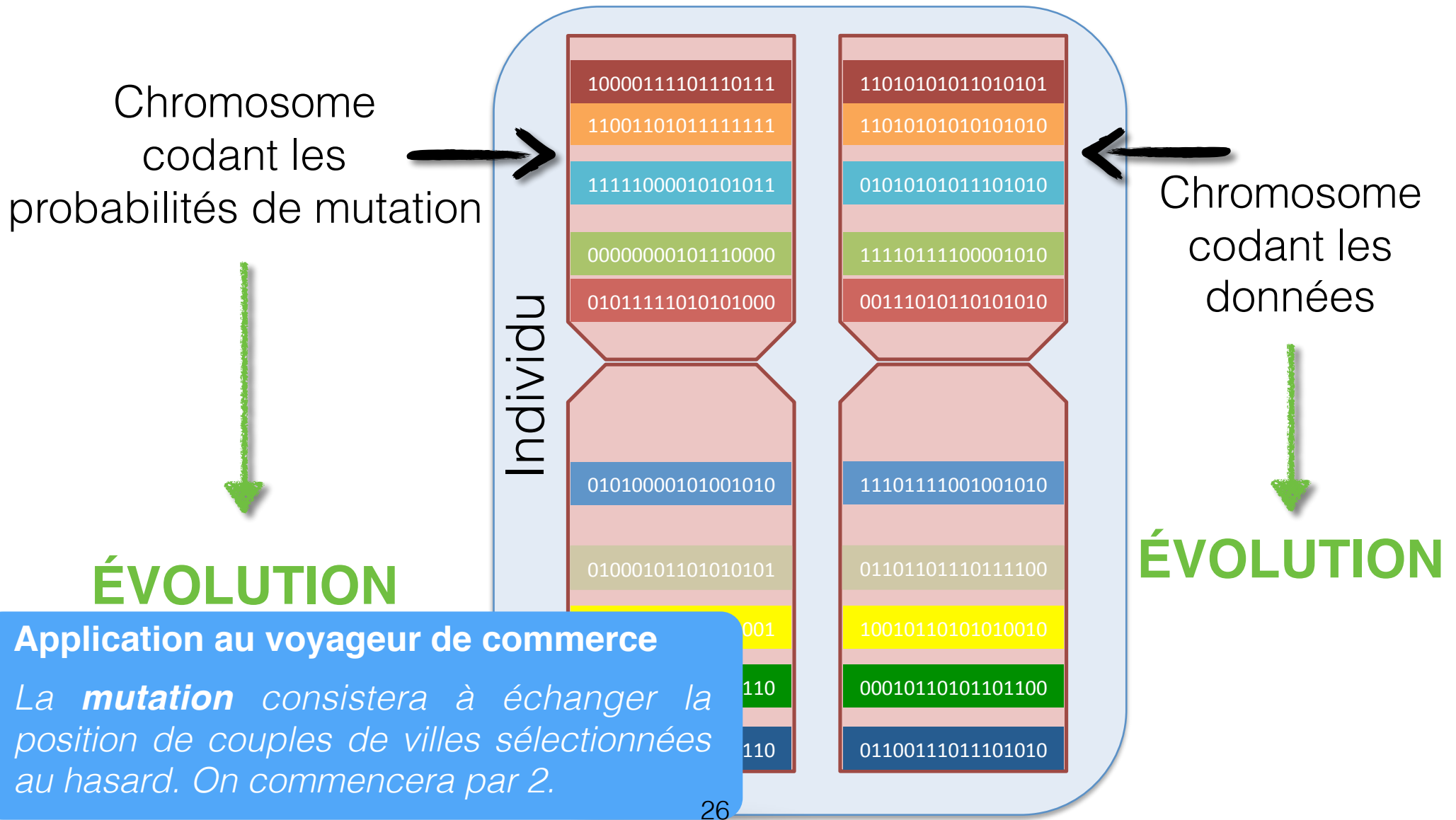
$$\Rightarrow p_m = 1/\mathcal{L} \text{ où } \mathcal{L} = \text{nb de bits du chromosome}$$

$$\Rightarrow p_m \text{ variable d'un gène à l'autre}$$

$$\Rightarrow p_m \text{ variable au cours du temps}$$

# ÉTAPE 5

## MUTATION AUTO-ADAPTATIVE



# Algorithmique génétique

Évolution

## Pratique et tâtonnement

Valeurs-clés

**Non-déterministe**

**Proche de l'optimal**