

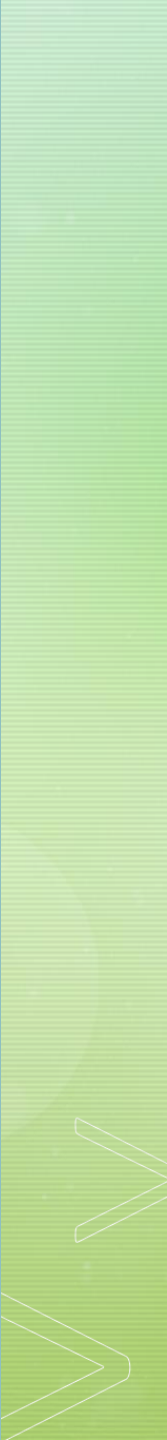
Christophe Rodrigues



# DataScience & IA



# Plan de la séance

- Déroulement général du cours
  - Introduction de l'IA
  - Algorithmes Evolutionnaires
- 



# Déroulement du cours

- 6 séances de cours (CM online - 1h15)
- 8 séances de Travaux Pratiques (3h)
- Évaluation : 1 projet final

# Déroulement du cours

- l'IA et son histoire
- Algorithmes évolutionnaires
- Algorithmes de recherche, règles d'associations, A\*
- Satisfaction de contraintes
- Problèmes avec opposants
- Machine Learning - méthodes de bases
- Machine Learning - méthodes avancées

# Intelligence Artificiel : Définitions

- " la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique " Marvin Lee Minsky
- IA forte
  - But : avoir toutes les apparences de l'intelligence humaine
  - Être capable de résoudre tout type de problème
  - Pour plus d'informations sur une IA générale: <https://agi.mit.edu/>
- IA faible
  - But: se concentre sur une tâche précise

# Optimisme des débuts de l'IA

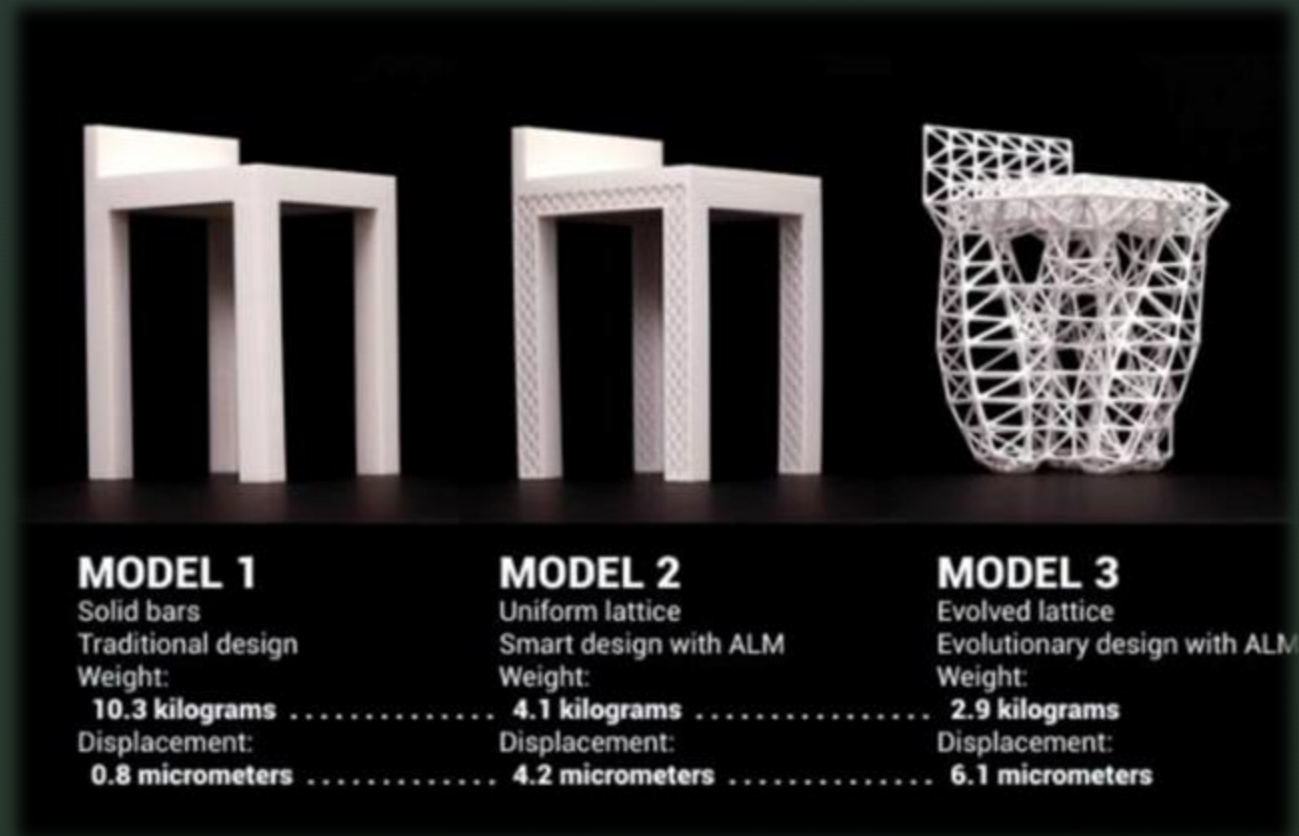
- H.Simon et Allen Newell en 1958 : « d'ici dix ans un ordinateur sera le champion du monde des échecs » et « d'ici dix ans, un ordinateur découvrira et résoudra un nouveau théorème mathématique majeur ». (Deep Blue battu Kasparov en 1997)
- H.Simon en 1965: « des machines seront capables, d'ici vingt ans, de faire tout travail que l'homme peut faire »



# Algorithmes Evolutionnaires

## Exemples d'applications

- Automated design - Project Dreamcatcher (Autodesk)



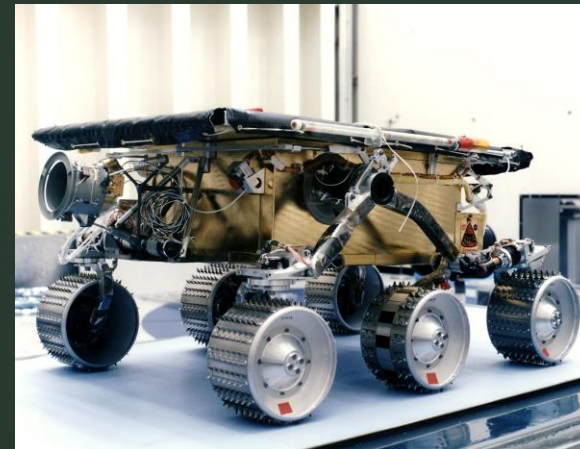
# Algorithmes Evolutionnaires

## Exemples d'applications



Robot aïbo de Sony  
Apprentissage de la marche

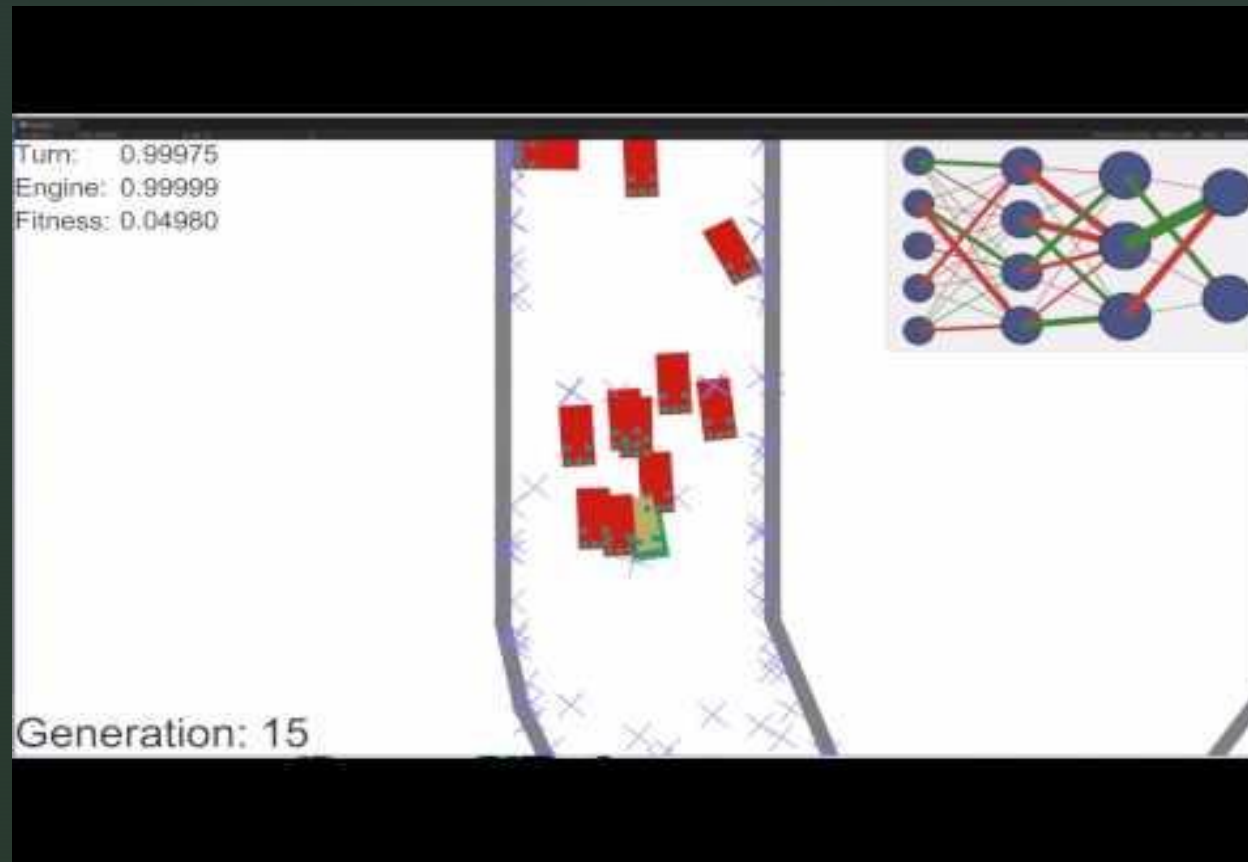
Déplacements du robot  
Pathfinder envoyé sur Mars





# Algorithmes Evolutionnaires

## Exemples d'applications



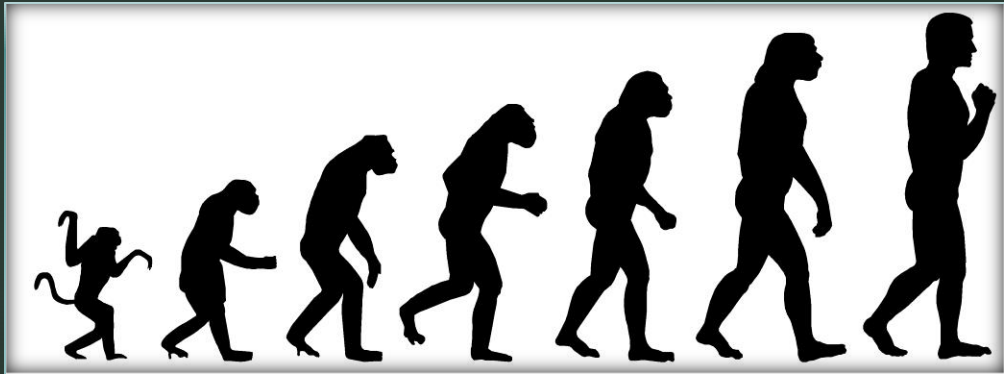
[https://github.com/ArztSamuel/Applying\\_EANNsadd](https://github.com/ArztSamuel/Applying_EANNsadd)

# Algorithmes Evolutionnaires

## Exemples d'applications

- Une centaine de domaines d'applications différents:
  - Finances
  - Informatiques
  - Industrie
  - Biologie/Bio-informatique
  - Conception
  - ...
- [https://en.wikipedia.org/wiki/List\\_of\\_genetic\\_algorithm\\_applications](https://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications)

# Algorithmes Evolutionnaires



- Inspiré par la théorie de l'évolution de Darwin
- Introduit par J. Holland en 1975

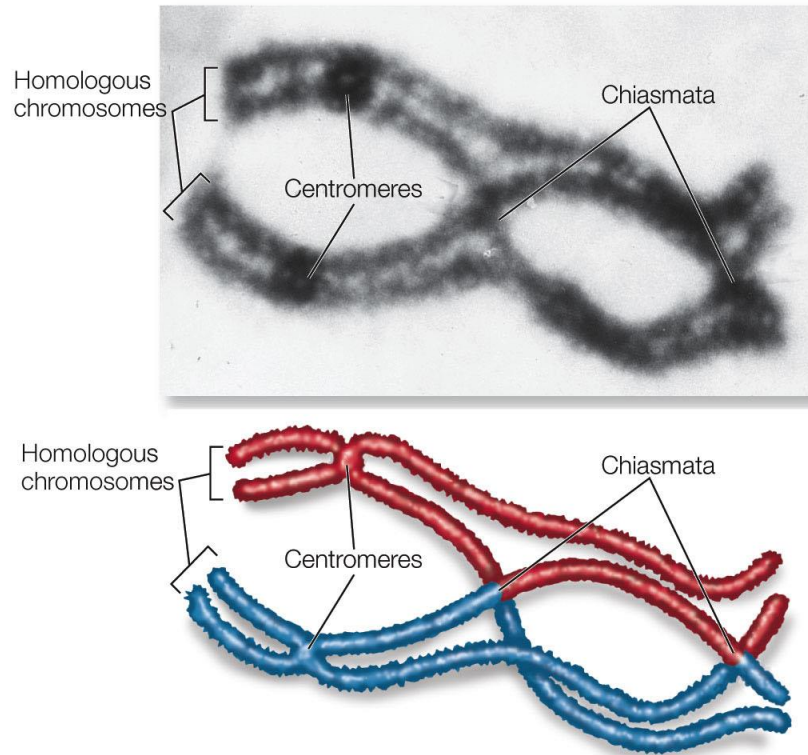
# Inspiration Biologique

- A partir d'organismes simples
  - vers des organismes de plus en plus complexes
- Transmission du patrimoine génétique des parents
  - Gène : porteur de l'information

# Inspiration Biologique

- L'évolution entraîne l'adaptation de la population
  - génération après génération
- Mécanisme en oeuvre:
  - Brassage génétique (hérédité)
  - Mutation génétique (aléatoire)
  - Sélection naturelle

# Brassage génétique : cross-over



**LIFE 8e, Figure 9.17**

LIFE: THE SCIENCE OF BIOLOGY, Eighth Edition © 2007 Sinauer Associates, Inc. and W. H. Freeman & Co.

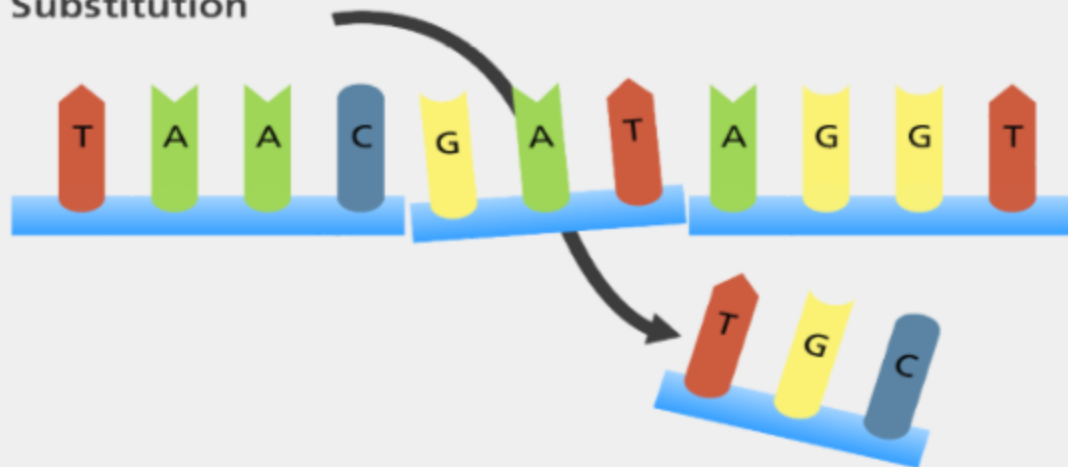


# Mutation génétique - Substitution

Original sequence



Substitution

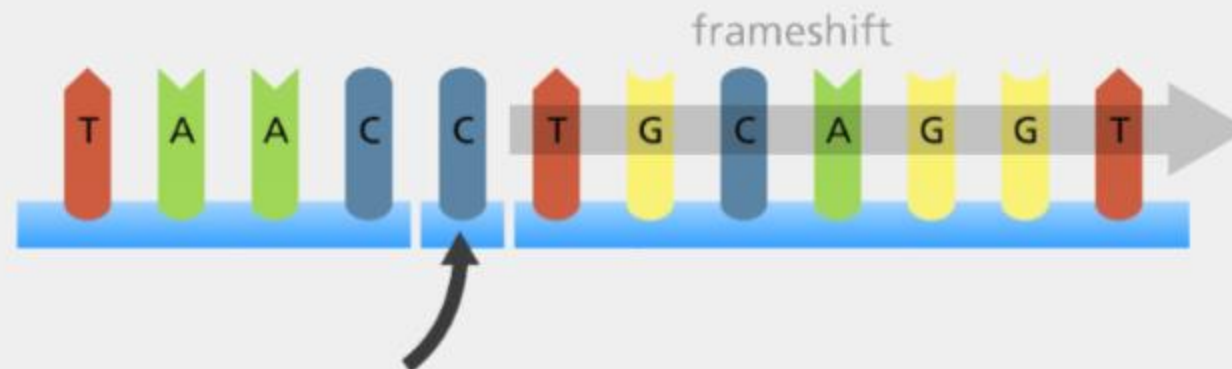


# Mutation génétique - Insertion

Original sequence



Insertion

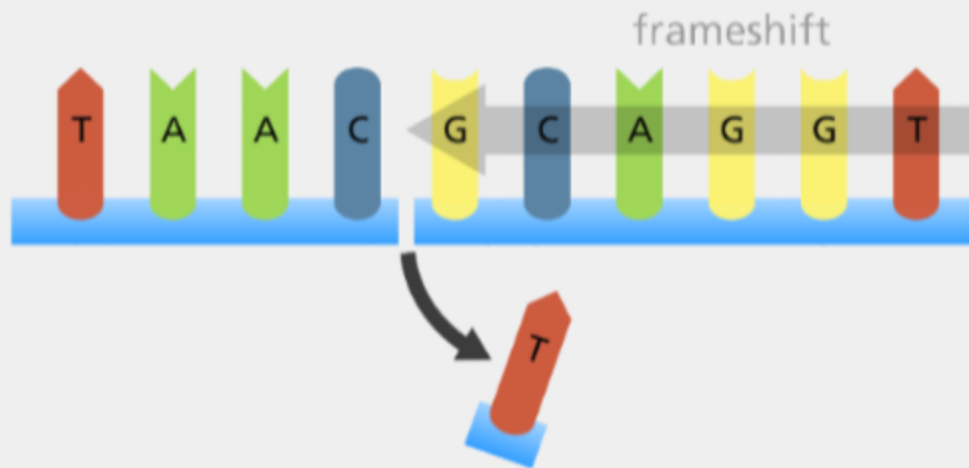


# Mutation génétique - Délétion

Original sequence



Deletion



# Sélection naturelle

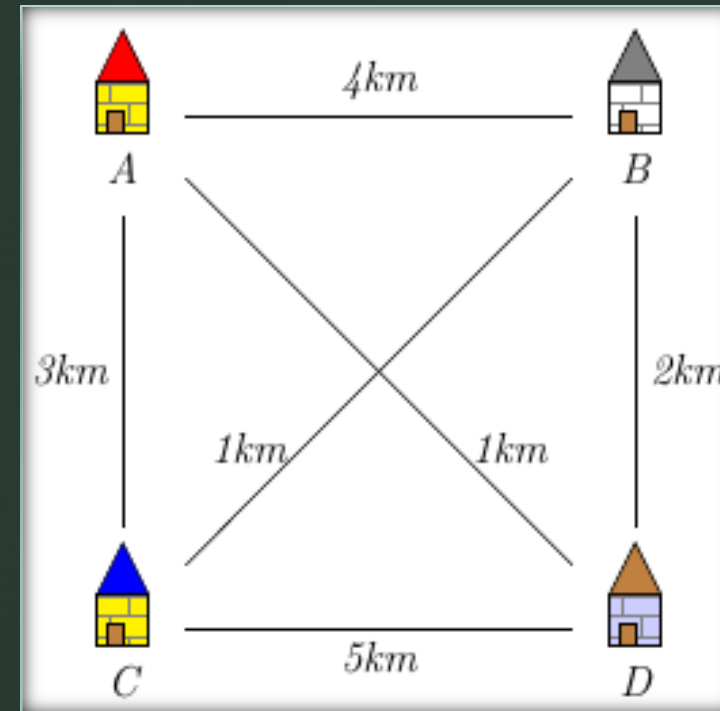
- Exemple : le coup des girafes



# Algorithme évolutionnaire

## exemple : voyageur de commerce

- But : voyageur doit passer par toutes les villes une fois en parcourant le moins de distance possible.
- Une solution possible (individu) : ABDC
- 20 villes :  $6 \times 10^{16}$  chemins



# Opérateur de croisement

- Une solution possible (individu) : ABDC

- Opérateur de croisement d'individus:

A partir de **ABCD** et **ACDB**

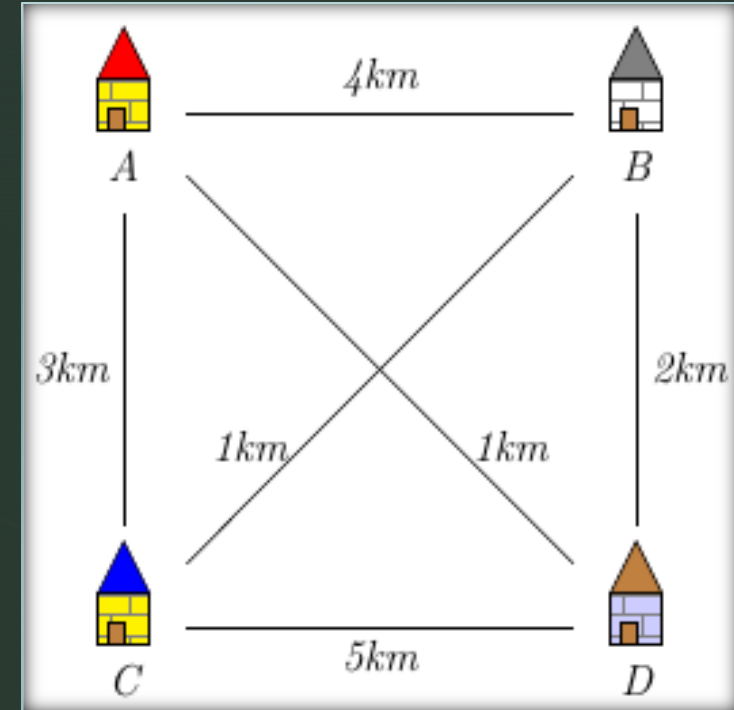
On obtient **ABDB** et **ACCD**

Cependant ces derniers ne sont pas corrects

Des villes apparaissent deux fois!

-> les répétitions sont remplacées par une ville au hasard non visitée.

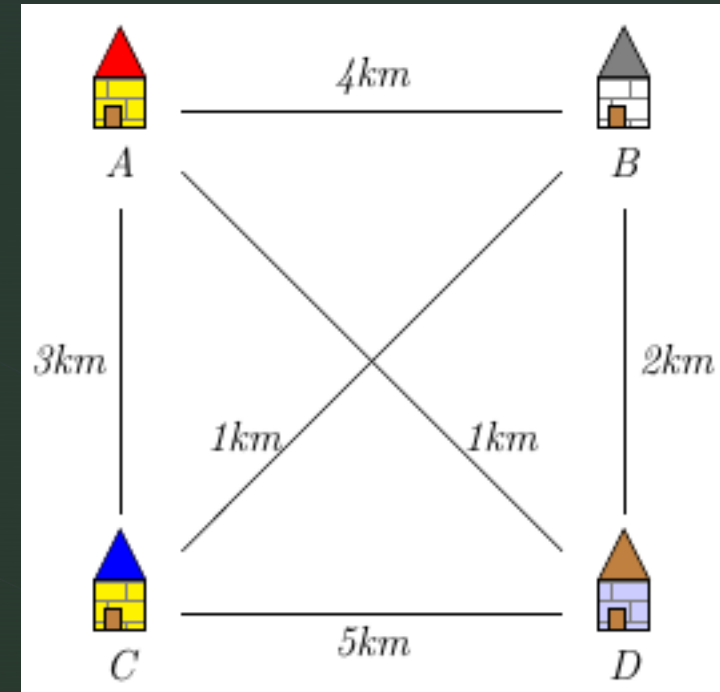
On obtient finalement deux nouveaux individus valides : ABDC et ACBD





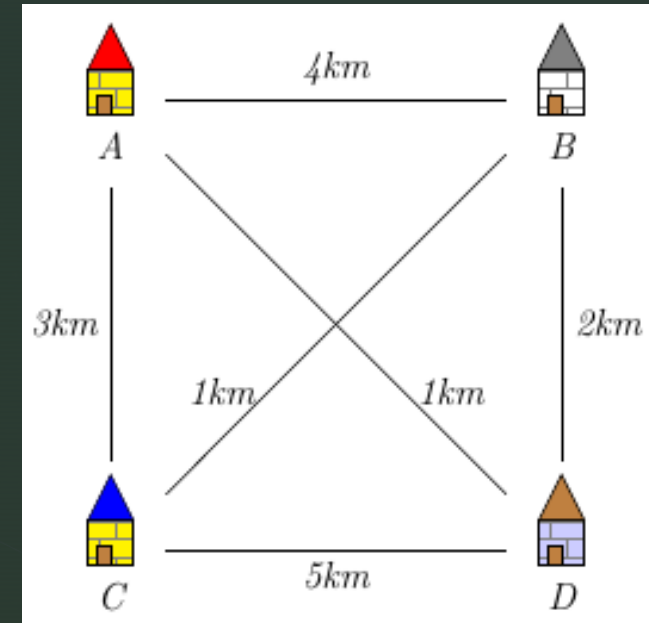
# Opérateur de mutation

- Aléatoirement, une ville est permutée avec une autre
  - $ABCD \rightarrow ADCB$



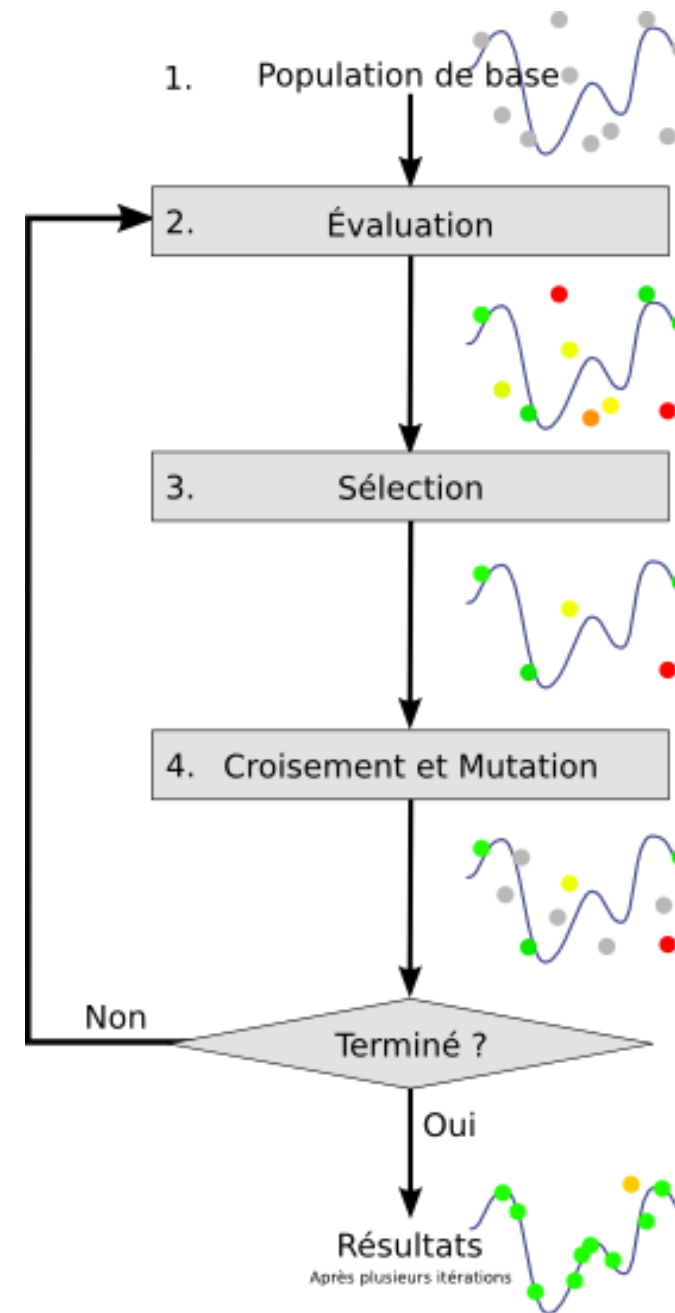
# Sélection

- Fonction fitness : distance parcourue
- $ABDC = 14 \text{ km}$ ,  $ADBC = 7 \text{ km}$
- Sélection :
  - Proportionnelle à la performance
  - Proportionnelle au rang (lissage)
  - Par tournoi (parallélise et élimine des comparaisons)
    - on garde le meilleur de chaque tournoi

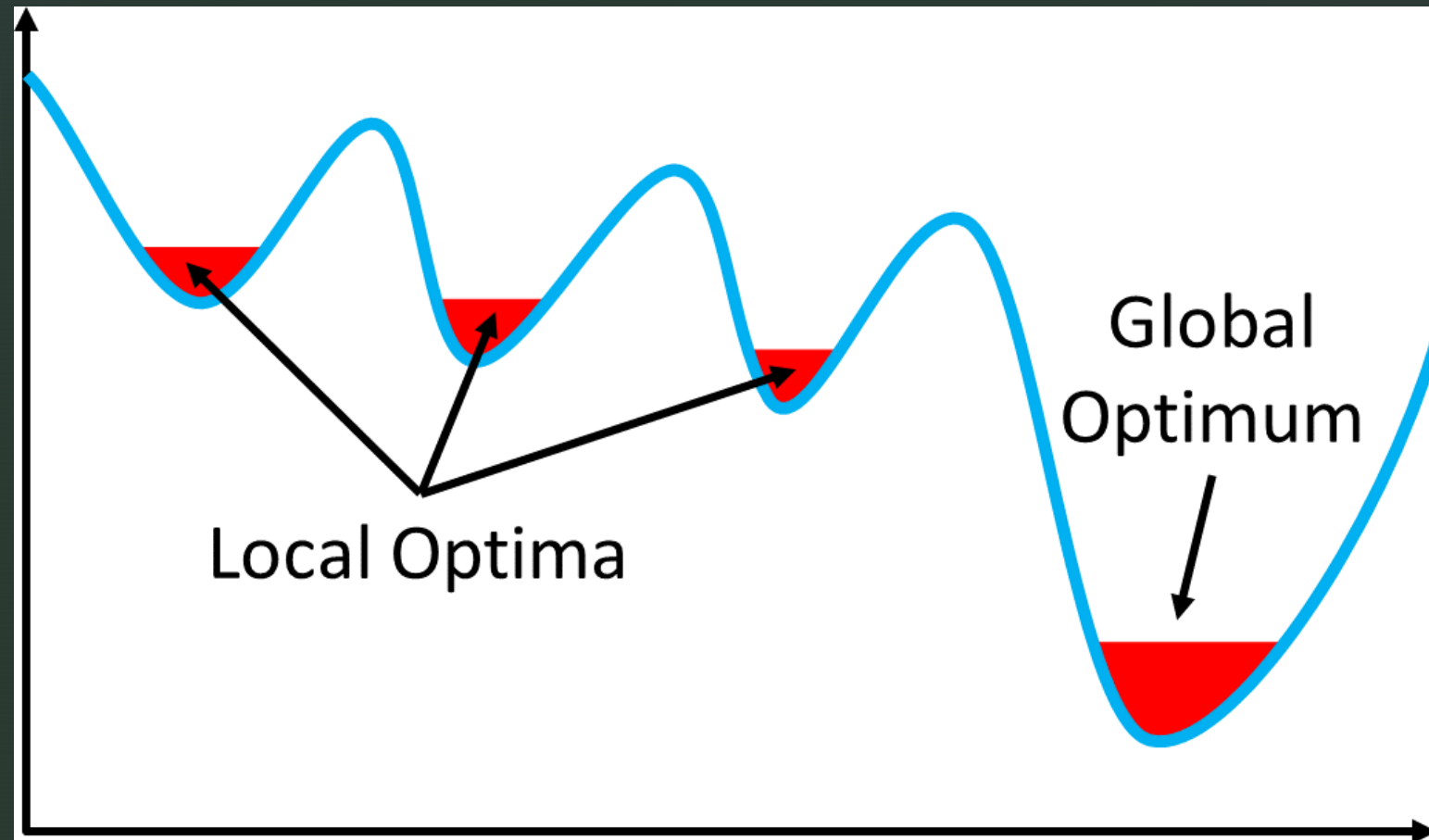


## Boucle générale

- La première population est constituée d'individus aléatoires.
- Afin de garantir une diversité de la population seule une fraction des meilleurs remplace deux de la génération précédente



# Optimum Local



# Algorithmes évolutionnaires et optimums locaux

- En théorie : une population ne peut tomber dans un optimum local indéfiniment si des individus aléatoirement sélectionnés sont ajoutés à chaque génération.
- En pratique : On considère qu'une population a convergé s'il n'y a plus d'amélioration après quelques générations. On peut relancer plusieurs populations et vérifier (ou non) la stabilité des meilleurs individus obtenus.
- Solution nécessitant beaucoup de temps de calcul mais à condition d'avoir une mesure d'évaluation(fitness) permet de traiter beaucoup de problèmes