# Optimisation numérique et Stratégie d'évolution

## Recherche Opérationnelle et Optimisation
## Master 1

Sébastien Verel
verel@lisic.univ-littoral.fr
http://www-lisic.univ-littoral.fr/~verel

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe CAMOME

# Optimisation numérique

**Définition : Problème d'optimisation numérique**

- Espace de recherche : ensemble de toutes les solutions possibles,

$$\mathcal{X} \subset \mathbb{R}^n$$

- Fonction objectif : critère de cout (minimisation)

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

**But : Résoudre un problème d'optimisation numérique**

Trouver une des meilleures solution selon le critère :

$$x^\star = \mathrm{argmin}\ f$$

*Mais, des fois, ensembles des meilleures solutions, bonne approximation de la meilleure solution, bonne solution 'robuste', etc.*

# Espace de dimension $n$

**Activité :**

Découverte de $\mathbb{R}^n$ avec la feuille TP 05 partie 1 (exercice 1)

# Bibliographie


**Bibliographie**

Daniele A. Di Pietro, Université de Montpellier,
Cours master 1, optimisation numérique :
`http://www.math.univ-montp2.fr/~di-pietro/Teaching.html`

# Direction de descente

**Definition : direction de descente**

Soit $\mathcal{X} \subset \mathbb{R}^n$, $f : \mathcal{X} \to \mathbb{R}$, et $x \in \mathcal{X}$.

On dit que $w \in \mathcal{X} \setminus \{0\}$ est une direction de descente en $x$ s'il existe un réel $\sigma_0 > 0$ tel que :

$$\forall \sigma \in [0, \sigma_0], \quad f(x + \sigma\ w) \leqslant f(x)$$

# Direction de descente

**Definition : direction de descente**

Soit $\mathcal{X} \subset \mathbb{R}^n$, $f : \mathcal{X} \to \mathbb{R}$, et $x \in \mathcal{X}$.

On dit que $w \in \mathcal{X} \setminus \{0\}$ est une direction de descente en $x$ s'il existe un réel $\sigma_0 > 0$ tel que :

$$\forall \sigma \in [0, \sigma_0], \quad f(x + \sigma\ w) \leqslant f(x)$$

**Definition : direction de descente stricte**

Soit $\mathcal{X} \subset \mathbb{R}^n$, $f : \mathcal{X} \to \mathbb{R}$, et $x \in \mathcal{X}$.

On dit que $w \in \mathcal{X} \setminus \{0\}$ est une direction de descente stricte en $x$ s'il existe un réel $\sigma_0 > 0$ tel que :

$$\forall \sigma \in [0, \sigma_0], \quad f(x + \sigma\ w) < f(x)$$

# Méthode de descente

**Algorithme de descente**

Choisir solution initiale $x \in \mathcal{X}$

**repeat**

  Trouver une dir. de descente stricte en $x$ : $w \in \mathcal{X} \setminus \{0\}$

  Choisir un nombre réel $\sigma > 0$

  $x \leftarrow x + \sigma\ w$

**until** critère d'arrêt non verifié

# Méthode de descente

**Algorithme de descente**

Choisir solution initiale $x \in \mathcal{X}$
**repeat**
  Trouver une dir. de descente stricte en $x$ : $w \in \mathcal{X} \setminus \{0\}$
  Choisir un nombre réel $\sigma > 0$
  $x \leftarrow x + \sigma \; w$
**until** critère d'arrêt non verifié

Questions légitimes :

- Comment choisir la direction de descente $w$ en fonction de $x$ ?
- Comment choisir le step size $\sigma$ ?
- Comment définir le critère d'arrêt ?

# Descente du gradient

**Activité :**

Découverte du gradient $\mathbb{R}^n$ avec la feuille TP 05 partie 1

# Direction du gradient

## Intuitions

Informellement (ou physiquement),
le **gradient** indique le vecteur vitesse de la surface (trajectoire),
c.-à-d. la direction, le sens et l'amplitude de la vitesse de
croissance de la surface.

## Définition formelle

Si $f$ est différentiable en $x \in \mathbb{R}^n$,
le gradient de $f$ en $x$ est égale à :

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \dots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

# Direction du gradient

## Résultat formel

Soit $f$ une fonction continûment dérivable sur un ouvert contenant $x \in \mathbb{R}$. On note par . le produit scalaire sur $\mathbb{R}^n$.

Si $w \in \mathcal{X} \setminus \{0\}$ est une direction de descente, alors $\nabla f(x).w \leqslant 0$.

Si $\nabla f(x) \neq 0$, alors $w = -\nabla f(x)$ est une direction de descente stricte en $x$.

# Méthode de la descente du gradient

**Algorithme de descente du gradient**

Choisir solution initiale $x \in \mathcal{X}$
**repeat**
  $w \leftarrow -\nabla f(x)$
  Choisir un nombre réel $\sigma > 0$
  $x \leftarrow x + \sigma \, w$
**until** critère d'arrêt non verifié

# Méthode de la descente du gradient

**Algorithme de descente du gradient**

Choisir solution initiale $x \in \mathcal{X}$
**repeat**
  $w \leftarrow -\nabla f(x)$
  Choisir un nombre réel $\sigma > 0$
  $x \leftarrow x + \sigma\ w$
**until** critère d'arrêt non verifié

Questions légitimes :

- Comment choisir le step size $\sigma$ en fonction de $f$ et $x$ ?
- Comment définir le critère d'arrêt ?

# Descente de gradient à pas fixe

**Algorithme : descente de gradient à pas fixe**

Choisir la taille du pas $\sigma \in \mathbb{R}^+$
Choisir solution initiale $x \in \mathcal{X}$
**repeat**
 $w \leftarrow -\nabla f(x)$
 $x \leftarrow x + \sigma\ w$
**until** critère d'arrêt non verifié

# Descente de gradient à pas fixe

**Algorithme : descente de gradient à pas fixe**

Choisir la taille du pas $\sigma \in \mathbb{R}^+$
Choisir solution initiale $x \in \mathcal{X}$
**repeat**
  $w \leftarrow -\nabla f(x)$
  $x \leftarrow x + \sigma \; w$
**until** critère d'arrêt non verifié

Questions légitimes :

- Comment choisir le step size $\sigma$ ?
- Comment définir le critère d'arrêt ?

# Méthode de Newton

**Activité :**

Découverte de la méthode de Newton avec la feuille TP 05 partie 1 (exercice 3)

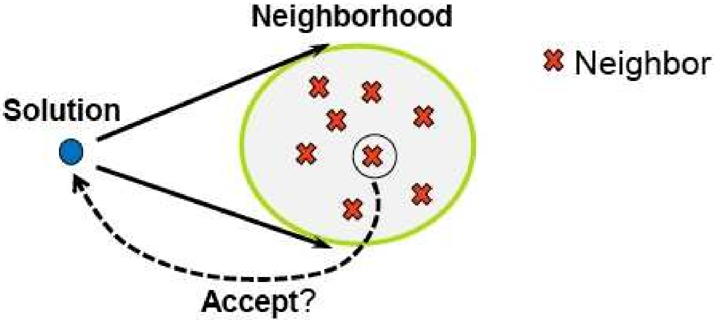
# Introduction aux stratégie d'évolution (evolution strategy)

**Cours introductif de l'école d'été en évolution artificielle**

- Anne Auger, juin 2012 : https://sites.google.com/site/ecoleea2012/programme

# Stochatic algorithms with unique solution (Local Search)

- $\mathcal{S}$ set of solutions (search space)
- $f : \mathcal{S} \to \mathbb{R}$ objective function
- $\mathcal{V}(s)$ set of neighbor's solutions of $s$



# Recherche Locale (LS)

- $\mathcal{S}$ ensemble des solutions (espace de recherche),
- $f : S \to \mathbb{R}$ fonction objectif à maximiser (ou coût à minimiser)
- $\mathcal{V}(s)$ ensemble des solutions voisines de $s$

*Algorithme d'une Recherche Locale*

Choisir solution initiale $s \in \mathcal{S}$
**repeat**
  choisir $s^{'} \in \mathcal{V}(s)$
  **if** accept(s, $s^{'}$) **then**
    $s \leftarrow s^{'}$
  **end if**
**until** critère d'arrêt non verifié

# Hill-Climber (HC)

Heuristique d'**exploitation** maximale.

*Hill Climber (best-improvement)*

Choisir solution initiale $s \in \mathcal{S}$
**repeat**
  Choisir $s^{'} \in \mathcal{V}(s)$ telle que $f(s^{'})$ est minimale
  $s \leftarrow s^{'}$
**until** $s$ optimum local

- Algorithme de comparaison
- Opérateur local de base de métaheuristique

# Optimum local / global

## Optimum local

Etant donné $(\mathcal{S}, f, \mathcal{V})$, $f$ à minimiser.

$x^\star$ est un optimum local ssi pour tout $x \in \mathcal{V}(x^\star)$, $f(x^\star) \leqslant f(x)$

## Optimum local strict

Etant donné $(\mathcal{S}, f, \mathcal{V})$, $f$ à minimiser

$x^\star$ est un optimum local ssi pour tout $x \in \mathcal{V}(x^\star)$, $f(x^\star) < f(x)$

## Optimum global

Etant donné $(\mathcal{S}, f, \mathcal{V})$, $f$ à minimiser.

$x^\star$ est un optimum global ssi pour tout $x \in \mathcal{S}$, $f(x^\star) \leqslant f(x)$

# Hill-Climbing first-improvement

*Hill-climber First-improvement (minimizer)*

Choisir solution initiale $s \in \mathcal{S}$
**repeat**
  Choisir $s^{'} \in \mathcal{V}(s)$ aléatoirement
  **if** $f(s^{'}) \leq f(s)$ **then**
    $s \leftarrow s^{'}$
  **end if**
**until** $s$ optimum local OU nbr d'éval. $\leq$ maxNbEval

# $(1+1)$-Evolution Strategy

**$(1+1)$-ES**

Choose randomly initial mean $m \in \mathbb{R}^n$
**repeat**
  $x' \leftarrow m + \sigma\, \mathcal{N}(0, C)$
  **if** $f(x')$ is better than $f(m)$ **then**
    $m \leftarrow x'$
  **end if**
**until** critère d'arrêt non verifié

$\sigma \in \mathbb{R}$ (step size) et la matrice $C \in \mathbb{R}^{n \times n}$ (covariance matrix) sont des paramètres de l'algorithme.

# $(1+1)$-Evolution Strategy with One-fifth success rule

**$(1+1)$-Evolution Strategy with $1/5$ rule**

Choose randomly initial solution $m \in \mathbb{R}^n$
**repeat**
  $x' \leftarrow m + \sigma\, \mathcal{N}(0, C)$
  **if** $x'$ is better than $m$ **then**
    $m \leftarrow x'$
    $\sigma \leftarrow \sigma \times exp(1/3)$
  **else**
    $\sigma \leftarrow \sigma / exp(1/3)^{1/4}$
  **end if**
**until** critère d'arrêt non verifié

La matrice $C \in \mathbb{R}^{n \times n}$ (covariance matrix) est un paramètre de l'algorithme.

# $(\mu/\mu, \lambda)$-Evolution Strategy

**$(\mu/\mu, \lambda)$-ES**

Choose randomly initial mean $m \in \mathbb{R}^n$
**repeat**
  **for** $i \in \{1 \ldots \lambda\}$ **do**
    $x_i^{'} \leftarrow m + \sigma\, \mathcal{N}(0, C)$
  **end for**
  Select the $\mu$ best solutions from $\{x_1^{'}, \ldots, x_\lambda^{'}\}$
  Let be $x_{:j}$ those solutions ranking by increasing order of $f$ :
    $f(x_{:1}) \leq \ldots \leq f(x_{:\mu})$

  $m \leftarrow \sum_{j=1}^{\mu} w_j x_{:j}^{'}$
**until** critère d'arrêt non verifié

avec $\hat{w}_i = \log(\mu + 0.5) - \log(i)$ et $w_i = \hat{w}_i / \sum_{i=1}^{\mu} \hat{w}_i$