Stratégie d'évolution

Recherche Opérationnelle et Optimisation
Master 1 I2L

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

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe CAMOME

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`

Optimization 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)

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

But : Résoudre un problème d'optimisation numérique

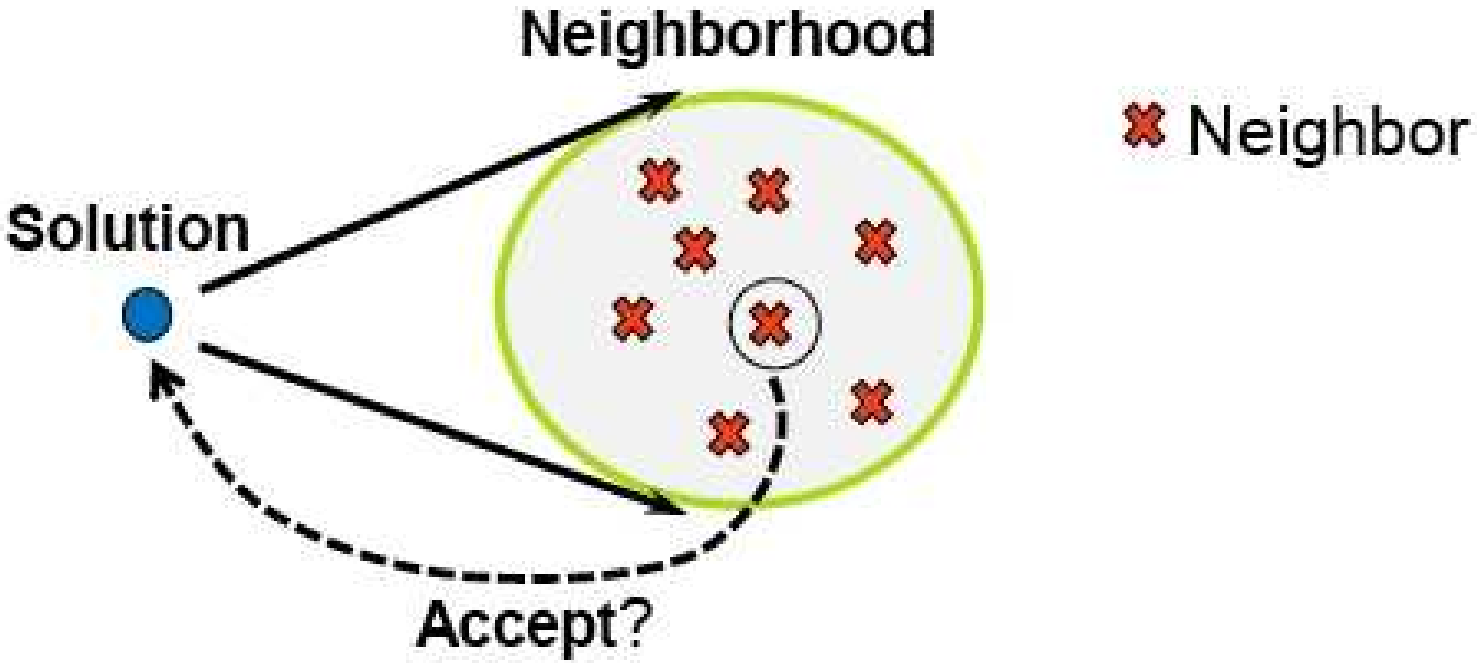
Trouver une des meilleures solution selon le critère :

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

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

Stochatic algorithms with unique solution (Local Search)

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



Recherche Locale (LS)

- $\mathcal{S}$ ensemble des solutions (espace de recherche),
- $f : S \rightarrow \mathbb{R}$ 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 $s^{'} \in \mathcal{V}(s)$
  $s \leftarrow s^{'}$
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

Optimum local

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

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

Optimum local strict

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

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

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 solution $s \in \mathbb{R}^n$
repeat
  $s^{'} \leftarrow s + \sigma\, \mathcal{N}(0, C)$
  if $s^{'}$ is better than s **then**
    $s \leftarrow s^{'}$
  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.

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

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

Choose randomly μ initial solutions $s_i \in \mathbb{R}^n$
repeat
  for $i \in \{1 \ldots \lambda\}$ **do**
    $s_i^{'} \leftarrow m + \sigma\, \mathcal{N}(0, C)$
  end for
  Select the μ best solutions from $\{s_1^{'}, \ldots, s_\lambda^{'}\}$
  Let be $s_{:j}$ those solutions ranking by increasing order of f :
    $f(s_{:1}) \leq \ldots \leq f(s_{:\mu})$

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

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

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

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

Choose randomly initial solution $s \in \mathbb{R}^n$
repeat
  $s^{'} \leftarrow s + \sigma\, \mathcal{N}(0, C)$
  if s is better than $s^{'}$ **then**
    $s \leftarrow s^{'}$
    $\sigma \leftarrow \sigma \times exp(1/3)$
  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.