May 27th 2015

Numerical Optimization: Basic Concepts and Algorithms

R. Duvigneau

Outline

- Some basic concepts in optimization
- Some classical descent algorithms
- Some (less classical) semi-deterministic approaches
- Illustrations on various analytical problems
- Constrained optimality
- Some algorithm to account for constraints

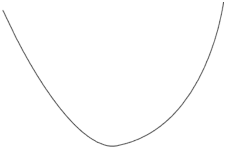
Some basic concepts

Problem description

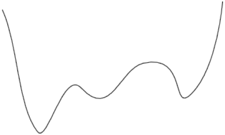
Definition of a single-criterion parametric problem with real unknown

$$
\begin{array}{llll}
\text{Minimize} & f(x) & x \in \mathbb{R}^n & \textit{cost fonction} \\
\text{Submitted to} & g_i(x) = 0 & i = 1, \cdots, l & \textit{equality constraints} \\
 & h_j(x) \geqslant 0 & j = 1, \cdots, m & \textit{inequality constraints}
\end{array}
$$

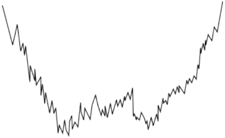
What does your cost function look like ?



Convex problem



Multi-modal problem



Noisy problem

Some commonly used algorithms

- Descent methods : adapted to **convex** cost functions
  steepest descent, conjugate gradient, quasi-Newton, Newton, etc.
- Evolutionary methods : adapted to **multi-modal** cost functions
  genetic algorithms, evolution strategies, particle swarm, ant colony, simulated annealing, etc.
- Pattern search methods : adapted to **noisy** cost functions
  Nelder-Mead simplex, Torczon's multidirectional search, etc.

Optimality conditions

Definition of a minimum

$x^\star$ is a minimum of $f : \mathbb{R}^n \mapsto \mathbb{R}$ if and only if there exists $\rho > 0$ such as:

- f defined on $\mathcal{B}(x^\star, \rho)$
- $f(x^\star) < f(y) \quad \forall y \in \mathcal{B}(x^\star, \rho) \quad y \neq x^\star$

$\rightarrow$ **not very useful to build algorithms ...**

Characterization

A sufficient condition for $x^\star$ to be a minimum is (if f twice differentiable):

- $\nabla f(x^\star) = 0$ (stationarity of gradient vector)
- $\nabla^2 f(x^\star) > 0$ (Hessian matrix positive definite)

Some classical descent algorithms

Model algorithm

For each iteration k (starting from x_k):

- Evaluate gradient $\nabla f(x_k)$
- Define a search direction $d_k(\nabla f(x_k))$
- Line search : choice of step length ρ_k
- Update: $x_{k+1} = x_k + \rho_k d_k$

Choice of the search direction

Steepest-descent method:

- $d_k = -\nabla f(x_k)$
- Descent condition ensured :
  $\nabla f(x_k) \cdot d_k = -\nabla f(x_k) \cdot \nabla f(x_k) < 0$
- But this yields an oscillatory path:
  $d_{k+1} \cdot d_k = (-\nabla f(x_{k+1})) \cdot d_k = 0$ (if exact line search)
- Linear convergence rate:
  $\lim_{k\to\infty} \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} = a > 0$

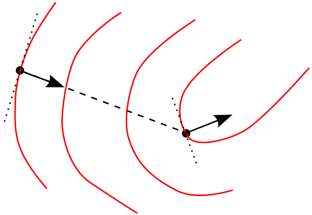


Illustration of steepest-descent path

Choice of the search direction

quasi-Newton method

- $d_k = -H_k^{-1} \cdot \nabla f(x_k)$ où H_k approximate of the Hessian matrix $\nabla^2 f(x_k)$
- H should fulfill the following conditions:
  - Symmetry
  - Positive definite: $\nabla f(x_k) \cdot d_k = -\nabla f(x_k) \cdot H^{-1} \cdot \nabla f(x_k) < 0$
  - 1D approximation of the curvature: $H_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)$
- Ex : BFGS method $H_{k+1} = H_k - \frac{1}{s_k^T H_k s_k} H_k s_k s_k^T H_k^T + \frac{1}{y_k^T s_k} y_k y_k^T$ où $s_k = x_{k+1} - x_k$ et $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
- Super-linear convergence rate : $\lim_{k \to \infty} \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} = 0$

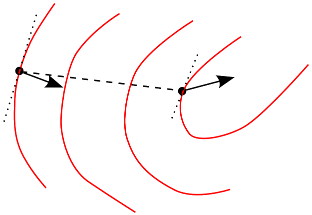


Illustration of quasi-Newton method

Choice of the step length

A classical criterion to ensure convergence : **Armijo-Goldstein**

- $f(x_k + \rho_k d_k) < f(x_k) + \alpha \nabla f(x_k) \cdot \rho_k d_k$ (Armijo)
- $f(x_k + \rho_k d_k) > f(x_k) + \beta \nabla f(x_k) \cdot \rho_k d_k$ (Goldstein)

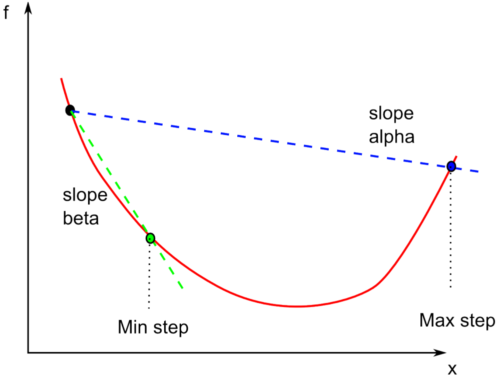


Illustration of Armijo-Goldstein criterion

Choice of the step length

An other criterion to ensure convergence (gradient required) : **Armijo-Wolfe**

- $f(x_k + \rho_k d_k) < f(x_k) + \alpha \nabla f(x_k) \cdot \rho_k d_k$ (Armijo)
- $\nabla f(x_k + \rho_k d_k) \cdot d_k > \beta \nabla f(x_k) \cdot d_k$ (Wolfe)

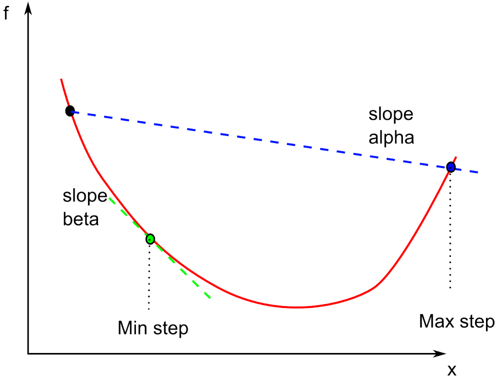


Illustration of Armijo-Wolfe criterion

Choice of the step length

The step length is determined using an **iterative 1D search**:

- Start from an initial guess $\rho_k^{(p)}$ $(p = 0)$
- Update to $\rho_k^{(p+1)}$:
  - Bisection method
  - Polynomial interpolation
  - ...
- until stopping criteria are fulfilled

A balance is necessary between the computational cost and the accuracy

Some (less classical) semi-deterministic approaches

Evolutionary algorithms

Principles

Inspired by Darwinian theory of evolution :

- A population is composed of individuals who have different characteristics
- Most fitted individuals can survive and reproduce
- An offspring population is generated from survivors

$\rightarrow$ **Mechanisms to improve progressively the population performance !**

Model algorithm (λ, μ)-ES

At each iteration k, a population is characterized by its mean $\bar{x}_k$ and its variance $\bar{\sigma}_k^2$.

Generation of population $k+1$:

- Generation of λ perturbation amplitudes $\sigma_i = \bar{\sigma}_k e^{\tau \; N(0,1)}$
- Generation of λ new individuals $x_i = \bar{x}_k + \sigma_i \; N(0, Id)$ (**mutation**) with $N(0, Id)$ multi-variate normal distribution
- Evaluation of the fitness of the λ individuals
- Choice of μ survivors among the λ new individuals (**selection**)
- Update of the population characteristics (**crossover** et **self-adaptation**) :

$$\bar{x}_{k+1} = \frac{1}{\mu} \sum_{i=1}^{\mu} x_i \qquad \bar{\sigma}_{k+1} = \frac{1}{\mu} \sum_{i=1}^{\mu} \sigma_i$$

Evolution strategy

Some results

- Proof of convergence **towards the global optimum** in a statistical sense :
  $\forall \epsilon > 0 \quad \lim_{k\to\infty} P(|f(\bar{x}_k) - f(x^\star)| \leqslant \epsilon) = 1$
- Linear convergence rate
- Capability to avoid local optima
- Limited to a rather small number of parameters ($\mathcal{O}(10)$)

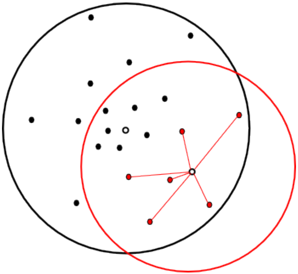


Illustration of evolution strategy step

Imprvement of ES algorithm by using an **anisotropic distribution**

- offspring population is generated using a covariance matrix C_k:

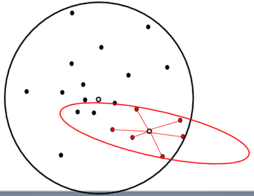
$$x_i = \bar{x}_k + \bar{\sigma}_k \; N(0, C_k) = \bar{x}_k + \bar{\sigma}_k \; B_k D_k N(0, Id)$$

avec B_k matrix of eigenvectors of $C_k^{1/2}$ et D_k eigenvalues matrix

- Iterative construction of the covariance matrix:

$$C_0 = Id \qquad C_{k+1} = \underbrace{(1-c)C_k}_{\text{previous estimation}} + \underbrace{\frac{c}{m} p_k p_k^T}_{\text{1D update}} + \underbrace{c(1-\frac{1}{m}) \sum_{i=1}^{\mu} \omega^i (y_i)(y_i)^T}_{\text{covariance of parents}} \text{ with :}$$

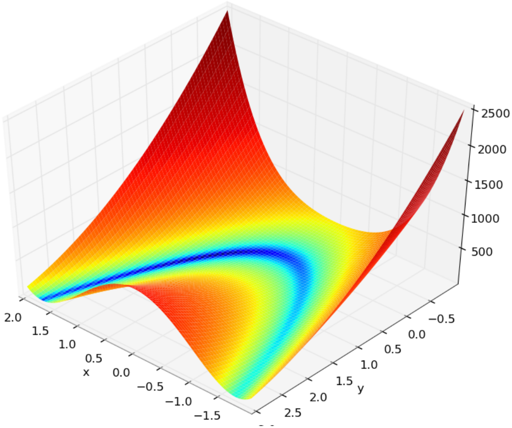
p_k evolution path (last moves) et $y_i = (x_i - \bar{x}_k)/\sigma_k$



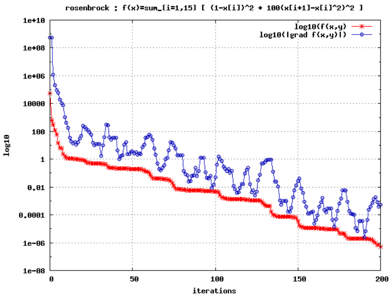
Some illustrations using analytical functions

Rosenbrock function

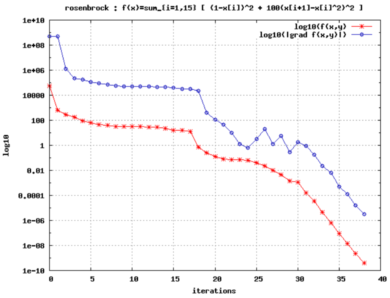
- Non-convex unimodal function "Banana valley"
- Dimension $n = 16$



Rosenbrock function

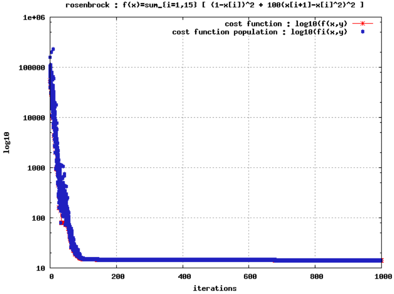


Steepest descent

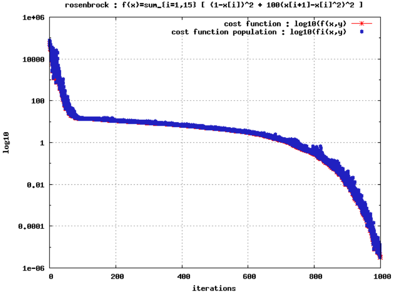


Quasi-Newton

Rosenbrock function



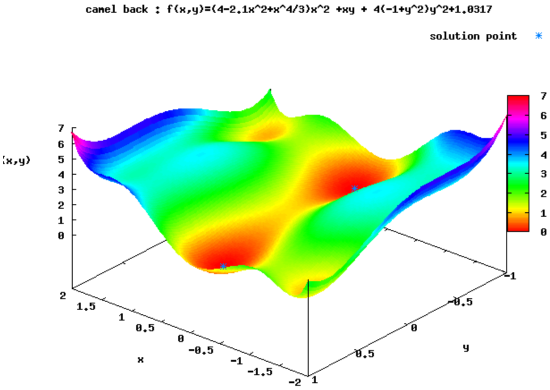
ES



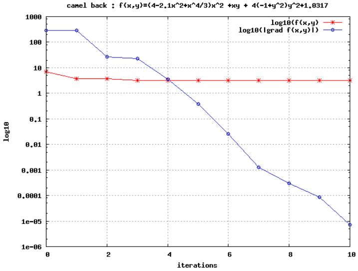
CMA-ES

Camelback function

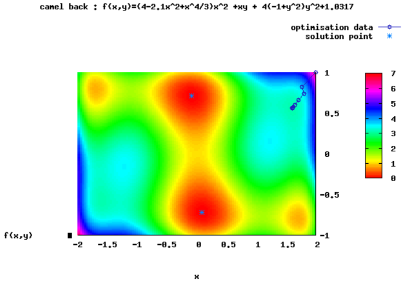
- Dimension $n = 2$
- Six local minima
- Two global minima



Camelback function

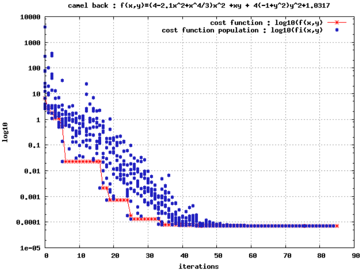


Quasi-Newton

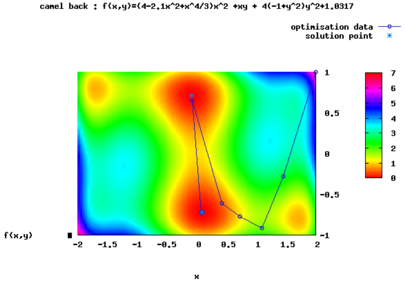


Optimization path

Camelback function



ES



Optimization path

Constrained optimality

Introduction

Necessity of constraints

- Often required to define a **well-posed problem from mathematical point of view** (existence, unicity)
- Often required to define a **problem that make sense from industrial point of view** (manufacturing)

Different types of constraints

- Equality / inequality constraints
- Linear / non-linear constraints

Linear contraints

Optimality conditions

A sufficient condition for $x^\star$ to be a minimum of f subject to $A \cdot x = b$:

- $A \cdot x^\star = b$ (admissibility)
- $\nabla f(x^\star) = \lambda^\star \cdot A$ with $\lambda^\star$ Lagrange multipliers (stationnarity)
- $A \cdot \nabla^2 f(x^\star) \cdot A > 0$ (projected Hessian positive definite)

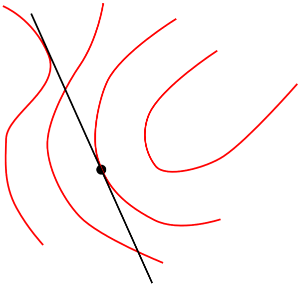


Illustration of optimality conditions for linear constraints

Linear constraints

Projection algorithm for descent methods

At each iteration k, from an admissible point x_k:

- Evaluation of gradient $\nabla f(x_k)$
- Choice of an **admissible search direction** $Z \cdot d_k$ with Z a projection matrix (in the admissible space: $A \cdot Z = 0$)
- Line search: choice of step length ρ_k
- Update : $x_{k+1} = x_k + \rho_k Z \cdot d_k$

Non-linear constraints

Optimality conditions

A sufficient condition for $x^\star$ to be a minimum of f subject to $c(x) = 0$:

- $c(x^\star) = 0$ (admissibility)
- $\nabla f(x^\star) = \lambda^\star \cdot A(x^\star)$ with $A(x) = \nabla c(x)$ (stationnarity)
- $A(x^\star) \cdot \nabla^2 \mathcal{L}(x^\star, \lambda^\star) \cdot A(x^\star) > 0$ with $\mathcal{L}(x, \lambda) = f(x) - \lambda \cdot c(x)$ (projected Lagrangian positive definite)

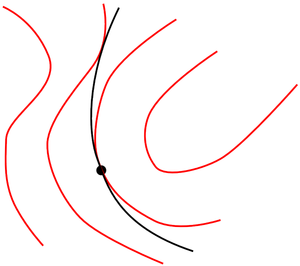


Illustration of optimality conditions for non-linear constraints

Non-linear constraints

Quadratic penalization algorithm

Cost function with penalization: $f_q(x,\kappa) = f(x) + \frac{\kappa}{2} c(x) \cdot c(x)$

It can be shown that: $\lim_{\kappa \to \infty} x^\star(\kappa) = x^\star$

Algorithm with quadratic penalization:

- Initialisation of κ
- Minimisation of $f_q(x,\kappa)$
- Increase κ to reduce constraint violation

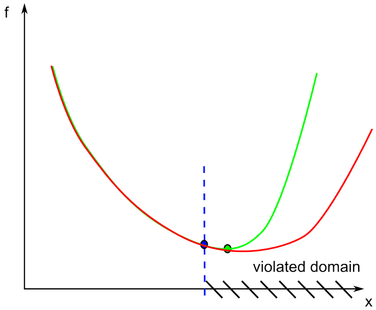


Illustration of quadratic penalization

Non-linear constraints

Absolute penalization algorithm

Cost function with penalization: $f_a(x,\kappa) = f(x) + \kappa \, \|c(x)\|$

It can be shown that: $\exists \kappa^\star$ such that $x^\star(\kappa) = x^\star \quad \forall \kappa > \kappa^\star$

Algorithm with absolute penalization :

- Initialisation of κ
- Minimisation of $f_a(x,\kappa)$
- Increase κ until constraint satisfied

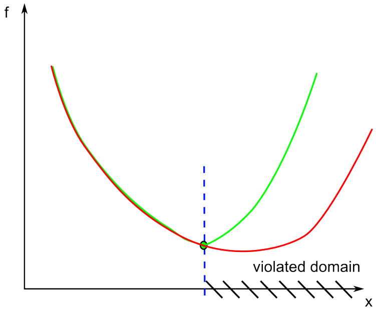


Illustration of absolute penalization

Non linear constraints

Optimality condition in terms of Lagrangian $\mathcal{L}(x,\lambda) = f(x) - \lambda \cdot c(x)$

- $\nabla_\lambda \mathcal{L}(x^\star,\lambda^\star) = 0$ (admissibility)
- $\nabla_x \mathcal{L}(x^\star,\lambda^\star) = 0$ (stationnarity)
- $A(x) \cdot \nabla^2 \mathcal{L}(x^\star,\lambda^\star) \cdot A(x) > 0$ (positive-definite)

SQP algorithm (Sequential Quadratic Programing)

At each iteration k, Newton method applied to (x,λ):

$$\begin{pmatrix} \nabla^2 f(x_k) - \lambda_k \cdot \nabla^2 c(x_k) & -A(x_k) \\ -A(x_k) & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta x \\ \delta \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k) + \lambda_k \cdot A(x_k) \\ c(x_k) \end{pmatrix}$$

Some references

Classical methods

- G. N. Venderplaats. Numerical optimization techniques for engineering design. McGraw-Hill, 1984.
- R. Fletcher. Practical Methods of Optimization. John Wiley & Sons, 1987.
- P. E. Gill, W. Murray, and M. H. Wright. Practical Optimization. Academic Press, 1981.

Evolutionary methods

- Z. Michalewics. Genetic algorithms + data structures = evolutionary programs. AI series. Springer-Verlag, New York, 1992.
- D. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Company Inc., 1989.