Optimization

1 Introduction

1.1 Objectif

At it's simplest, optimization is the selection of the best element from some set. This often include minimizing (or maximizing) a function (Objective Function, OF) which depend on an value to be found (input).

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) := \{\mathbf{x}^* \mid \forall \mathbf{x} : f(\mathbf{x}^*) \leq f(\mathbf{x})\}.$$

Optimization can be extended to with several functions and/or input as well as behind constraint to a domain.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \end{aligned}$$

where:

- $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized over the variable x,
- $g_i(\mathbf{x}) \leq 0$ are called inequality constraints,
- $h_i(\mathbf{x}) = 0$ are called equality constraints.

1.2 History and Development

Fermat and Lagrange found calculus-based formulas for identifying optima, while Newton and Gauss proposed iterative methods for moving towards an optimum.

Dantzig published the Simplex algorithm in 1947, and John von Neumann developed the theory of duality in the same year.

2 Calculus of variations

2.1 Definition

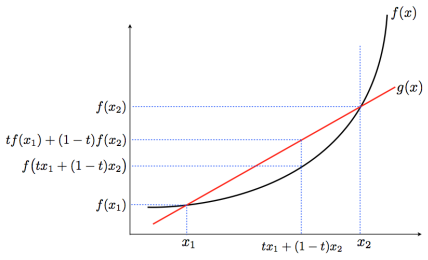
- A **stationary** point is where the derivative of the function is equal to zero ($f'(x_0) = 0$), it can be categorised in (with a small *epsilon* > 0):
  - **Extrema or turning point**
    - **Maximum**: $f'(x - \epsilon) > 0$ and $f'(x - \epsilon) < 0$
    - **Minimum**: $f'(x - \epsilon) < 0$ and $f'(x - \epsilon) > 0$
  - **Inflection**: change from concave to convex (or vis-versa)
    - **rising - :** $f'(x - \epsilon) > 0$ and $f'(x - \epsilon) > 0$
    - **falling - inflection**: $f'(x - \epsilon) < 0$ and $f'(x - \epsilon) < 0$
  - Saddle point
- A **monotonic** function is a function between ordered sets that preserves the given order. A **unimodal function** if for some value m, it is monotonically increasing for x m and monotonically decreasing for x m
- The **Hessian matrix** is a square matrix of second-order partial derivatives of a scalar-valued function $\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$.
- A **functional** is a function from a vector space into a scalar. With f being a linear function:

$$\begin{aligned} \text{function} \quad & x_0 \mapsto f(x_0) \\ \text{functional} \quad & f \mapsto f(x_0) \end{aligned}$$

- Implicational relationships between statements
  - **Nececessary**: "N if S" (N $\Leftarrow$ S)
  - **Sufficient**: "if S, then N" (S $\Rightarrow$ N)
  - **Necessary and sufficient**: "S if and only if N" (S $\Leftrightarrow$ N).
- A function is **convex** if the line between any two point on the graph lies above the graph. More formally, $f : X \to \mathbb{R}$ is a convex function if $\forall x_1, x_2 \in X, \forall t \in [0, 1]$:

$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$$

- In euclidean space, an object is convex if every point on a straight line joining any two point of the space is also within the object.



- The **convex hull** of a set X of points in the Euclidean plane or Euclidean space is the smallest convex set that contains X.
- Polygon is a plane figure (2D) that is bounded by a finite chain of straight line segments (edges or sides) meeting at vertices and closing in a loop to form a closed chain or circuit.
- A **polytope** is a geometric object with flat sides which generalise the 2D polygon in n-dimensional.
- Convex Polytope is a polytope whose point form a convex set. It can be written in equation form as $Ax \leq b$ which correspond to a linear programming constraint.

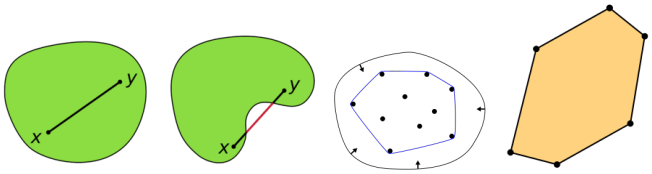


Figure 1: (a) convex set,(b) not convex set,(c) convex hull,(d) 2D-polytope

2.2 Fermat's theorem, Euler-Lagrange equation

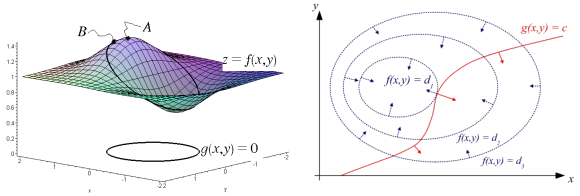
- **Fermat's theorem** states that at any point where a differentiable function attains a local extremum, its derivative is zero. Let $f\colon (a,b) \to \mathbb{R}$ be a differentiable function on an extremum $x_0 \in (a,b)$, then $f'(x_0) = 0$
- **Euler-Lagrange equation** is a partial differential equation whose solutions are the functions for which a given functional is stationary. In 1D,

$$\frac{\partial F}{\partial f} - \frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial F}{\partial f'} = 0$$

The idea behind the equation is that we want to find the function $f(x)$ which minimize (or maximize) a real-valued function $F\colon (x, f(x), f'(x)) \mapsto \mathbb{R}$. In order to find it, we set the derivative of the integral of F to zero, which end up in a partial derivative function known as the Euler-Lagrange equation.

2.3 Method of Lagrange multipliers

Lagrange method is used for first order derivable continuous function with only equality constrains ($h(\mathbf{x}) = 0$).



The idea is that at the solution ($\mathbf{x}^*$), the gradient vector of both function (f, h) will have the same direction:

$$\nabla f(\mathbf{x}^*) = -\lambda \cdot \nabla h(\mathbf{x}^*)$$

Where λ is a multiplier called the Lagrange multiplier. In order to find this point, we can use the Lagrange function :

$$\Lambda(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda \cdot h(\mathbf{x})$$

which will have a stationary point at the solution

$$\nabla_{\mathbf{x},\lambda}\Lambda(\mathbf{x}^*, \lambda^*) = 0$$

Expending this equation rise to $n+1$ equations with $n+1$ variable to solved. Note that $\nabla_\lambda \Lambda = 0$ force $h(\mathbf{x}) = 0$. Multiple constraint can be added and Karush-Kuhn-Tucker generalized it for inequality constraint

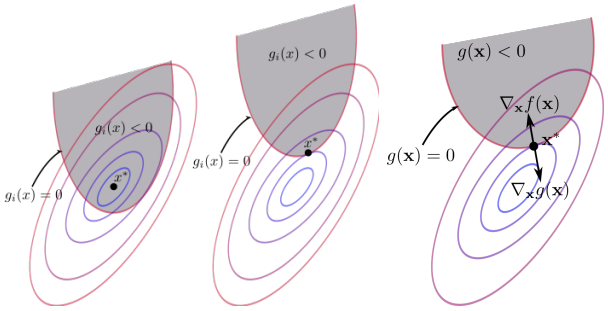
2.4 Karush-Kuhn-Tucker conditions

The KKT approach generalizes the method of Lagrange multipliers to inequality constraints.

$$\nabla f(\mathbf{x}^*) = -\sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) - \sum_{i=1}^{m} \mu_i \nabla g_i(\mathbf{x}^*)$$

For the inequality constraint, we can show that $\mu_i g_i(\mathbf{x}^*) = 0$ because $\mu_i = 0$ when the solution is outside the inequality space ($g_i(\mathbf{x}^*) < 0$, constraint null) and otherwise, $g_i(\mathbf{x}^* = 0$ because the solution is at the boarder of the space.

Because the solution on $g_i(\mathbf{x}) > 0$ or $g_i(\mathbf{x}) = 0$, μ_i can only be zero or greater than zero ($\nabla g_i(\mathbf{x})$ has opposite sign to $\nabla f(\mathbf{x})$)



2.5 Lagrange duality

Duality means that optimization problems may be viewed from either of two perspectives, the primal problem or the dual problem (the duality principle). The solution to the dual problem provides a lower bound to the solution of the primal (minimization) problem.

$$\Lambda(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}) + \sum_{i=1}^{m} \mu_i \nabla g_i(\mathbf{x})$$

$$\sum_{j=1}^{l} \lambda_j h_j(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i g_i(\mathbf{x}^*) \leq 0$$

and therefore the Lagrange dual function is the lowest bound of the primal problem:

$$g(\lambda, \mu) = \inf \Lambda(\mathbf{x}, \lambda, \mu) \leq \Lambda(\mathbf{x}^*, \lambda, \mu) \leq f(\mathbf{x}^*)$$

2.6 Markov-chain Monte Carlo (MCMC)

- Monte Carlo are broad class of algorithm that rely on repeated random sampling to obtain numerical results.
- Markov-chain is a random process that undergoes transitions from one state to another where the probability distribution of the new state depends only on the current state and not on the sequence of events that preceded it (Markov property).

$$\Pr(x_{n+1} \mid x_1, \ldots, x_n) = \Pr(x_{n+1} \mid X_n)$$

- MCMC combine both properties.

3 Problem Categorisation

3.1 Global vs Local

Global (or absolute) method intend to find the best estimate over the whole space rather than part of the parameter space (local or relative). Strict extremum is the only one best value.(

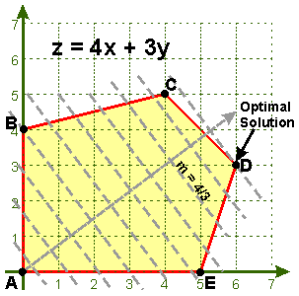
$$f(\mathbf{x}^*) \underbrace{\leq}_{< \text{ for strict}} f(\mathbf{x}^* + \epsilon) \qquad \begin{cases} \exists \epsilon > 0 & \text{local} \\ \forall \epsilon > 0 & \text{global} \end{cases}$$

3.2 Linear (LP), non-linear (NLP) and quadratic programming

LP is an convex optimization technique used with linear objective function, subject to linear equality and/or inequality constraints.

minimize	$f(\mathbf{x}) = \mathbf{c}^{\mathrm{T}}\mathbf{x}$
subject to	$A\mathbf{x} \leq \mathbf{b}$
and	$\mathbf{x} \geq \mathbf{0}$

These constraint correspond to a convex polytope. Because of the OF is linear (can always decrease in a certain direction), we know that it minimum occur at a boundary of the constraint polytope.



For quadratic problem, the OF is describe with:

$$f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$$

The constrain are usually the same as linear but could also be quadratic

3.3 Heuristic and Meta-heuristic

Heuristic method search for an approximate solution ("good enough") by trading optimality (the best), completeness (all), accuracy, or precision for speed. In order to find this approximate, informations about the problem are required. Meta-heuristic are called approximation (rather than approximate) in the sense that they are proven close solution, which can therefore be applied to any problem without adaptation.

3.4 Pattern search

Pattern search (also called Direc-Search) is a family of optimization that do not require the gradient of the problem.

3.5 Stochastic vs Deterministic

In stochastic optimization, unknown parameter are handle with uncertainty and pdf while

3.6 Multi-objective optimization and Pareto Front

For non-trivial multi-objective optimization problem, there does not exist a single solution that simultaneously optimizes each objective. A solution is called Pareto optimal (or efficient) if none of the objective functions can be improved without degrading some of the other objective values. Vector optimization is the generalized case.

3.7 Rate of Convergence

Suppose that the sequence x_k converges to the number L. Classification:

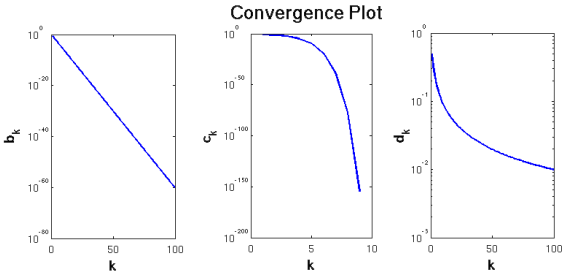
$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|} = \mu$$

- **Linear**, $\mu \in (0, 1)$
- **Superlinearly**, $\mu = 0$. We say that the sequence converges with order q to L for $q > 1$ if: ($q = 2$ is quadratic convergence)

$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^q} = \mu \quad \text{with} \mu > 0$$

- **Sublinearly**, $\mu = 1$. The sequence converges logarithmically to L if

$$\lim_{k\to\infty} \frac{|x_{k+2} - x_{k+1}|}{|x_{k+1} - x_k|} = 1$$



4 Optimization algorithms

4.1 Simplex Algorithm

Linear programing,
One of the top 10 algorithm of the twentieth century, Simplex algorithm solved linear programme (LP) using the property that the minimum is always at a vertex.

> Start on a vertex of the polytope
> **while** One of the neighbouring vertices has a lower OF **do**
> Move to the new vertex
> **end while**

Numerical solution for this involve simplex tableaux and pivot operations. The revised simplex method implement the same mathematical expression but in matrices form. Other method which uses this basic idea is the path-following algorithms.

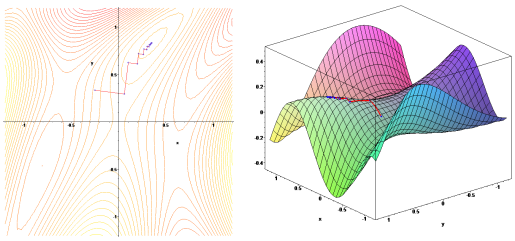
5 Gradient-Based Algorithm (Iterative methods)

5.1 Gradient descent

Gradient-based, differentiable
Gradient descent is an interative method which start with a point and move toward the negative gradient proportional to the step size (γ_n)

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla f(\mathbf{x}_n)$$

In some case, the optimal step size can be computed analytically.



5.2 Conjugate gradient method

linear or quadratic, gradient-based, iterative,
CGM is solving either an inverse problem for linear problem ($\mathbf{x}|\mathbf{Ax} = \mathbf{b}$) or an optimization problem for the quadratic expression whose gradient is the linear problem $\nabla f(\mathbf{x}) = \mathbf{Ax} - \mathbf{b}$. This is equivalent as the minimum of f will be found when his gradient is null.

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Ax} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$$

The matrix $\mathbf{A}$ need to be symmetric (i.e., $\mathbf{A}^T = \mathbf{A}$), positive definite (i.e. $\mathbf{x}^T\mathbf{Ax} > 0 \quad \forall \mathbf{x} \neq 0 \in \mathbb{R}^n$), and real (i.e. $\mathbf{A} \in \mathbb{R}^n$).

In the gradient descent, the iterative approach would use a direction of search $\mathbf{p}_n = \nabla f(\mathbf{x}) = -(\mathbf{A}\mathbf{x}_n - \mathbf{b})$ We can see that this expression is the residual of the linear problem denoted $\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \gamma_n(\mathbf{b} - \mathbf{A}\mathbf{x}_n)$$

But in CGM, the new direction of search need to be a conjugate gradient of the previous one $\langle \mathbf{p}_n, \mathbf{p}_k \rangle_A = 0, \quad \forall k < n$. This new direction can be found by subtraction the projection of the gradient to the previous direction:

$$\mathbf{p}_n = \mathbf{r}_n - \sum_{i<n} \operatorname{proj}_{\mathbf{p}_i}(\mathbf{r}_n) = \mathbf{r}_n - \sum_{i<n} \frac{\mathbf{p}_i^{\mathrm{T}} \mathbf{A}\mathbf{r}_n}{\mathbf{p}_i^{\mathrm{T}} \mathbf{A}\mathbf{p}_i}\mathbf{p}_i$$

And finally, this gave us:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n, \qquad \alpha_n = \frac{\mathbf{p}_n^{\mathrm{T}}\mathbf{b}}{\mathbf{p}_n^{\mathrm{T}}\mathbf{A}\mathbf{p}_n}$$

The advantage of Conjugate gradient method over Newton's method is that matrix is not store and therefore Conjugate Gradient Method is used when factorization is not feasible.

The biconjugate gradient method provides a generalization to non-symmetric matrices. Various nonlinear conjugate gradient methods seek minima of nonlinear equations.

5.3 Newton's Method

Newton's Method is a method for finding successively better approximations to the roots (or zeroes) of a real-valued differentiable function.

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{f(\mathbf{x}_n)}{f'(\mathbf{x}_n)}$$

5.3.1 Analysis

Newton's Method come from **Taylor Second order expension**

$$f(\mathbf{x}) = f(\mathbf{x}_n) + f'(\mathbf{x}_n)(\mathbf{x} - \mathbf{x}_n) + R_1$$

where the remainder is

$$R_1 = \frac{1}{2!} f''(\xi_n)(\mathbf{x} - \mathbf{x}_n)^2 \quad \text{with } \xi \in \mathbb{R}, |\mathbf{x} - \xi_n| < |\mathbf{x} - \mathbf{x}_n|$$

By setting $\epsilon_n = \mathbf{x} - \mathbf{x}_n$, and $f(\mathbf{x}) = 0$. we can find that under some condition the rate of convergence is quadratic[3.7]

$$\frac{|\epsilon_{n+1}|}{\epsilon_n^2} = \frac{|f''(\xi_n)|}{2|f'(\mathbf{x}_n)|}$$

5.3.2 Failure

Failure to converge can be caused by a stationary point ($f'(\mathbf{x}) = 0$), a poor initial estimate, overshooting or cyclic iteration.

If the derivative is unknown, the **secant method**, is a simplistic solution which replace the derivative with a backward finite difference approximation :

$$\mathbf{x}_n = \mathbf{x}_{n-1} - f(\mathbf{x}_{n-1}) \frac{\mathbf{x}_{n-1} - \mathbf{x}_{n-2}}{f(\mathbf{x}_{n-1}) - f(\mathbf{x}_{n-2})}$$

To avoid overshooting, a small step $\gamma_n \in (0, 1)$ (which can change at each iteration) can be add:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \frac{f(\mathbf{x}_n)}{f'(\mathbf{x}_n)}$$

5.3.3 Generalization

For a system of k non-linear equations $\mathbf{f}$ of k variable $\mathbf{x}$, the derivative $\mathbf{f}'$ is called the Jacobian matrix $\mathbf{J}_{i,j} = \frac{\partial f_i}{\partial x_j}$ and Newton's method become:

$$\mathbf{J}_f(\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{f}(\mathbf{x}_n)$$

For under-determined problem (number of equation smaller than number of variable), the generalized inverse **??** of the non-square Jacobian matrix $\mathbf{J}^+ = ((\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1})\mathbf{J}^{\mathrm{T}}$ instead of the inverse of $\mathbf{J}$.

5.4 Gauss-Newton Method

In over-determined problem, the Gauss-Newton method simplify the Newton method to minimize a function in a least square sense (instead of finding the root). In least-squre problem, the function to minimized is sum of square of the residual $\mathbf{r}(\mathbf{x})$ with m parameters $\mathbf{x}$

$$f\colon \mathbb{R}^m \to \mathbb{R} \qquad f(\mathbf{x}) = \sum_i \mathbf{r}_i^2(\mathbf{x})$$

$$\mathbf{J}_f\colon \mathbb{R}^m \to \mathbb{R}^m \qquad \mathbf{J}_f(\mathbf{x}) = 2\sum_i r_i(\mathbf{x}) \frac{\partial \mathbf{r}_i(\mathbf{x})}{\partial \mathbf{x}}$$

$$\mathbf{H}_f\colon \mathbb{R}^m \to \mathbb{R}^{m\times m} \qquad \mathbf{H}_f(\mathbf{x}) = 2\sum_i \left(\frac{\partial r_i}{\partial \beta_j}\frac{\partial r_i}{\partial \beta_k} + r_i \frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k} \right).$$

[...]

Starting from an optimization version of Newton method, where the function to minimized is the sum of square of the residual $f(\mathbf{x}) = \sum r_i^2$, where the residual is de difference between x and known data d, $r_i = x_i - d_i$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{f'(\mathbf{x}_n)}{f''(\mathbf{x}_n)}$$

where $f' = 2\sum r_i \frac{\partial r_i}{\partial \mathbf{x}}$ and $f'' = 2\sum r_i \frac{\partial r_i}{\partial \mathbf{x}}$ It can be shown that the Hessian can be approximate by $2\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + (\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f)^{-1}\mathbf{J}_f^{\mathrm{T}} f(\mathbf{x}_n)$$

We can observe ... The Jacobian became an orthogonal matrix which imply $\mathbf{J}_f^{\mathrm{T}} = \mathbf{J}_f^{-1}$ and $(\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f) = \mathbf{I}$. The Gaussiannetwon simplyfyied in the generalized solution of Newton.

This relationship can be found when looking at minima (stationary point, $f' = 0$).The Newton method is reformulate with the Hessian $\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j}$ replacing f by this jacobienne:

5.5 LevenbergMarquardt algorithm

The LMA interpolates between the GaussNewton algorithm (GNA) and the method of gradient descent by using a dumping factor λ (Levenberg):

$$\mathbf{x}_{n+1} = \mathbf{x}_n + (\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f + \lambda_n \mathbf{I})^{-1}\mathbf{J}_f^{\mathrm{T}} f(\mathbf{x}_n)$$

The (non-negative) damping factor, , is adjusted at each iteration. If reduction of S is rapid, a smaller value can be used, bringing the algorithm closer to the GaussNewton algorithm, whereas if an iteration gives insufficient reduction in the residual, can be increased, giving a step closer to the gradient descent direction. Marquardt improve this by weighting each dumping factor depending on the gradient (convergence) in each direction

$$\mathbf{x}_{n+1} = \mathbf{x}_n + (\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f + \lambda_n \mathrm{diag}(\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f))^{-1}\mathbf{J}_f^{\mathrm{T}}$$

The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as GaussNewton using a trust region approach.

5.6 Quasi-Newton Method

Strictly, any method that replaces the exact Jacobian with an approximation can is a quasi-newton method. But in practice, Variable Metric Methode is accumulating information to build up a better approximation of the inverse Hessian

6 Heuristics

6.1 Particle swarm-essaims de particule

meta-heuristique, not gradient based, no guarantee of global, Swarm intelligence, evolutionary computing
A set of S particles (candidate solutions) are moved around the search-space $[x_{min}, x_{max}]$ with its velocity (v_i^s) which is computed according to its current position (x_i^s), its best estimate (x_b^s) as well as the global best estimate (x_b^b).

for each particle $s = 0, \ldots, S$ **do**
  $x_0^s \leftarrow \mathcal{U}(x_{min}, x_{max})$ ▷ position
  $x_b^s \leftarrow x_0^s$ ▷ best position
  $v_0^s \leftarrow \mathcal{U}(-|x_{min}, x_{max}|, |x_{min}, x_{max}|)$ ▷ velocity
  if $f(x_0^s) < f(x_b^b)$ **then**
    $x_b^b \leftarrow x_0^s$
  end if
end for
while *criterion*$(i, f(x_b), f(x_{b,new})/f(x_{b,old}))$ **do**
  $i \leftarrow i + 1$
  for each particle $s = 0, \ldots, S$ **do**
    $r_b, r_{bb} \leftarrow \mathcal{U}(0, 1)$
    $v_i^s \leftarrow \omega v_{i-1}^s + \phi_b r_b (x_b^s - x_{i-1}^s) + \phi_{bb} r_{bb} (x_b^b - x_{i-1}^s)$
▷ update velocity
    $x_i^s \leftarrow x_{i-1}^s + v_i^s$ ▷ update position
    if $f(x_i^s) < f(x_b^s)$ **then**
      $x_b^s \leftarrow x_i^s$
      if $f(x_i^s) < f(x_b^b)$ **then**
        $x_b^b \leftarrow x_i^s$
      end if
    end if
  end for
end while

Method exist to compute best parameters$(r_b, r_{bb}, \omega, \phi_b, \phi_{bb})$.

6.2 Evolutionist algorithm

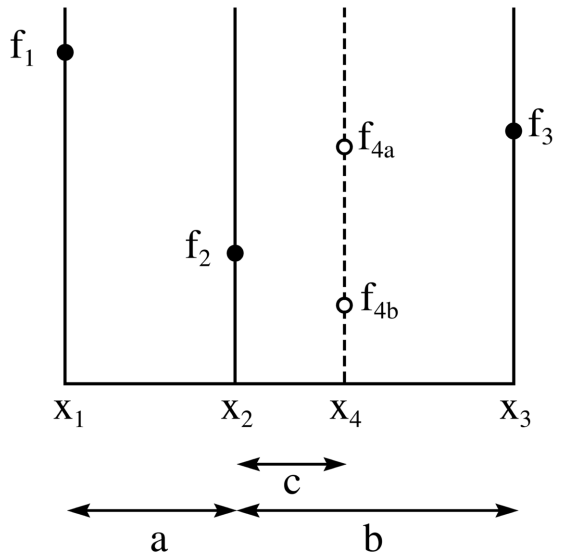
Population-based, stochastic, meta-heuristic
An evolutionist algorithm uses mechanisms inspired by biological evolution, such as reproduction, mutation, inheritance, crossover, and selection.

Generate randomly first generation
while criteria not met (time,fitness...) **do**
  - Evaluate the fitness of each individual
  - Select the best-fit individuals for reproduction
  - Breed new individuals through crossover and mutation operations to give birth to offspring
  - Replace least-fit population with new individuals
end while

Genetic algorithm and evolutionist strategie are different in the sence that recominbaison is randomized in genetic while evlutionist uses co-evolution in order to find convergence.
Read more : Schulze-Riegert and Ghedan, 2007

6.3 Golden Section Search

Unimodal function,
This method used for unimodal function successively narrows the range of values containing the solution by sampling function values for triples of points whose distances form a golden ratio. (see also Fibonacci search).



If the interval is divided by two (instead of forming a golden ratio), the method is called **Bisection method**.

6.4 Simulated annealing-recuit simul

meta-heuristic, global, stochastic

$x_b \leftarrow x_0$
while $criterion(i, f(x_b), f(x_{b,new})/f(x_{b,old}))$ **do**
  $x_i \leftarrow neighbour(x)$
  if $\Pr(f(x), f(x_i), i) > \mathcal{U}(0,1)$ **then**
    $x_b \leftarrow x_i$
  end if
end while

where:

- x_b is the current best estimate and x_i is the proposed new estimate
- $criterion()$ return true if none criteria is met otherwise false and stop the algorithm
- $neighbour()$ is a function returning a new proposed estimate from the current estimate. This is a sort of perturbation function.
- $\Pr(f(x), f(x_i), i)$ is the acceptance probabilities.
  - The probability increase with a increase in the difference of the OF of the new estimate :$(f(x) - f(x_i)) \nearrow, \Pr \nearrow$. ie: best estimate, best chance to be accepted
  - Usually, if $f(x) > f(x_i)$, $\Pr = 1$, ie: better estimate, sure to be accepted.
  - At the beginning, acceptance for worse estimate allow to explore more globally but along the iteration, it become more and more difficult to converge.

Method to find good neighbour and acceptance exist.