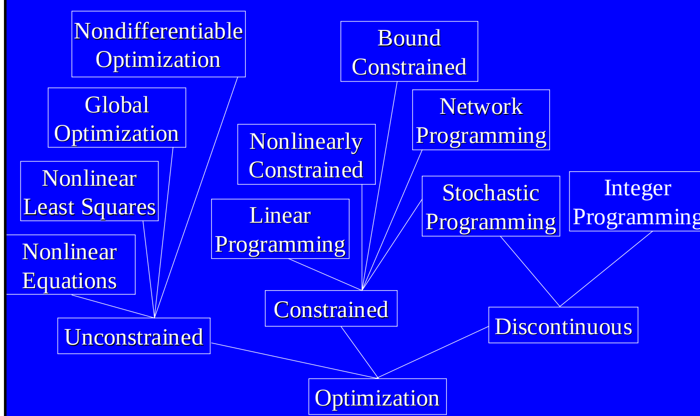
Mathematical Optimization

Optimization Taxonomy



Types of Minima

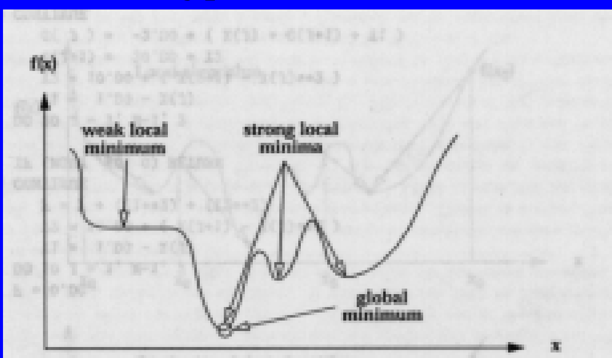


Figure 1: Types of Minima for Unconstrained Optimization Problems.

Local methods

1. Supply an initial guess x_0
2. For k = 0,1,2, … until convergence
3. Test x_k for convergence
4. Calculate a search direction p_k
5. Determine an appropriate step length α
6. Set $x_{k+1} = x_k + \alpha p_k$

Unconstrained optimization

Minimize f(x)

Gradient Descent

$$f(x_k + p)$$
$$p = -g_k$$

Conjugate Gradient

$$\min \frac{1}{2} x^T A x + b^T x$$

Is equivalent to

$$Ax = -b$$

$$r = -(Ax + b)$$
$$r_k^T d_j = 0 \forall j < k$$

Solution obtained in at most n steps.

Convergence

$x_k = \{x_0, x_1, ...\}$ converges to x_* if

$$\lim_{k \to \infty} |x_k - x_*| = 0$$
$$|x_{k+1} - x_*| \le c |x_k - x_*|$$

Superlinear Convergence

$$|x_{k+j} - x_*| \le c_k |x_k - x_*|$$

P-order convergence

$$|x_{k+j} - x_*| \le c |x_k - x_*|^p$$

Preconditioning

Turn $Ax = -b$

into

$$M^{-\frac{1}{2}} A M^{-\frac{1}{2}} \left(M^{\frac{1}{2}} x \right) = -M^{-\frac{1}{2}} b$$

M must be factored very rapidly

$$M^{-\frac{1}{2}} A M^{-\frac{1}{2}}$$

Has the same eigen-values as

$$M^{-1} A$$

Newton's Method

$$f(x_k + p) = f(x_k) + g_k^{\,T} p + \frac{1}{2} p^T H_k p$$

$$g_k^{\,T} + H_k p = 0$$

$$H_k p = -g_k^{\,T}$$

$$x_{k+1} = x_k + p$$

Truncated Newton's method

Newton's method is inaccurate far away from the solution, so don't compute exactly

Advance only as far as the following condition is satisfied:

$$\left\| g_k^{\,T} + H_k p \right\| \le n_k \left\| g_k \right\|$$

Quasi Newton Methods

Build an approximation to the Hessian B_k

$$B_{k+1} s_k = y_k$$

$$B_{k+1} = B_k - \frac{B_k s_k (B_k s_k)^T}{s_k^{\,T} B_k s_k} + \frac{y_k y_k^{\,T}}{y_k^{\,T} s_k} + \ldots$$

$$B_k p_k = -g_k$$

$$H_k = B_k^{-1}$$

$$H_{k+1} = H_k + \left(I - \frac{s_k y_k^{\,T}}{y_k^{\,T} s_k} \right) H_k \left(I - \frac{y_k s_k^{\,T}}{y_k^{\,T} s_k} \right) + \frac{s_k s_k^{\,T}}{y_k^{\,T} s_k}$$

$$p_k = -H_k g_k$$

Limited Memory

Don’t keep and update the dense matrix Hk

Instead, keep the last i (typically 5) values for y and p and update at each time step

$$H_{k+1} = H_k + \left(I - \frac{s_k y_k^T}{y_k^T s_k} \right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right) + \frac{s_k s_k^T}{y_k^T s_k}$$

$$p_k = -H_k g_k$$

Line Search

How far to go in the given direction?

Use quadratic or cubic interpolation of the univariate function

$$\phi(\alpha) = f(x_k + \alpha p_k)$$

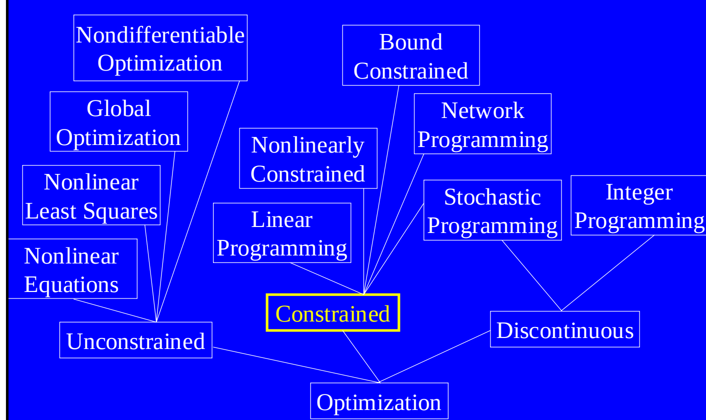
In search for suitable α

Trust Region

Keep track of how much you trust the accuracy of your update:

$$\|D_k p\|_2 < \Delta_k$$

Optimization Taxonomy



Constrained Optimization

Extend the unconstrained optimization ideas to handle constraints:

- Sequential Quadratic Programming (SQP)
- Augmented Lagrangian Methods
- Reduced Gradient Methods
- Feasible SQP

Constrained Minima

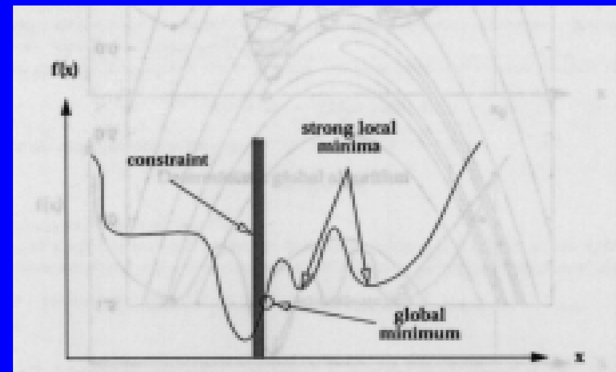


Figure 2: Types of Minima for Constrained Optimization Problems.

Lagrangian formulation

- Given $\underset{\mathbf{q}}{\text{minimize}} \quad G(\mathbf{q})$

  subject to $\mathbf{C}(\mathbf{q}) = 0$

- We define a Lagrangian $L(\mathbf{q},\boldsymbol{\lambda}) = G(\mathbf{q}) - \boldsymbol{\lambda}\cdot\mathbf{C}$

$$\underset{\mathbf{q},\boldsymbol{\lambda}}{\text{minimize}} \quad G(\mathbf{q}) - \boldsymbol{\lambda}\cdot\mathbf{C}$$

Newton's method and the Lagrangian

$$L(x_k + p) = L(x_k) + g_k^{\,T} p + \frac{1}{2} p^T H_k p$$

$$g_k^{\,T} + H_k p = 0$$

$$H_k p = -g_k^{\,T}$$

$$x_{k+1} = x_k + p$$

Solving the Lagrangian

- To solve $\frac{\partial G(\mathbf{q}) - \boldsymbol{\lambda}\cdot\mathbf{C}}{\partial\{\mathbf{q},\boldsymbol{\lambda}\}} = \mathbf{0}$ iteratively
- We setup the linear system

$$\begin{bmatrix} \frac{\partial^2 \mathbf{G}}{\partial^2 \mathbf{q}} & \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^T \\ \frac{\partial \mathbf{C}}{\partial \mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} d\mathbf{q} \\ d\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{G}}{\partial \mathbf{q}} - \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^T \boldsymbol{\lambda} \\ -\mathbf{C} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{q}_{new} \\ \boldsymbol{\lambda}_{new} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} + \alpha \begin{bmatrix} d\mathbf{q} \\ d\boldsymbol{\lambda} \end{bmatrix}$$

SQP

1. Pretend constraints are linear
2. Pretend objective is quadratic
3. Solve the quadratic subproblem using the Lagrangian formulation
4. Follow the subproblem direction
5. Goto step 1 until convergence

Augmented Lagrangian

Start adding things to the Lagrangian

$$L(\mathbf{q},\boldsymbol{\lambda},\nu) = G(\mathbf{q}) + \boldsymbol{\lambda}\cdot\mathbf{C} + \sum_i \nu_i C_i^2$$

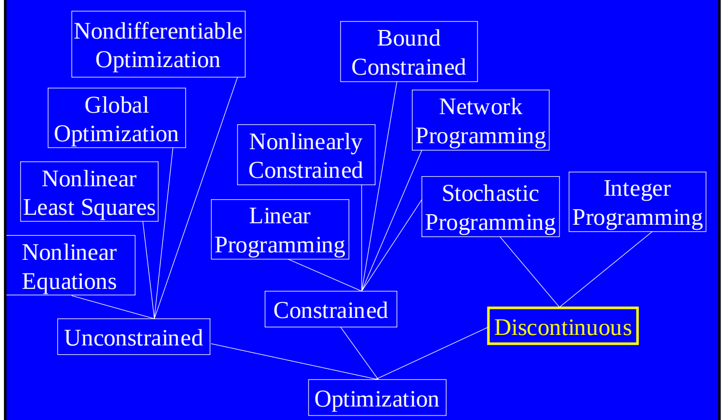
Feasible SQP

Useful when f(x) cannot be evaluated outside of the feasible region

$$\min f(x)$$
$$C(x) \le 0$$

Considerably slower than SQP

Optimization Taxonomy



Methods of Last Resort

- Simulated Annealing
- Genetic Algorithms