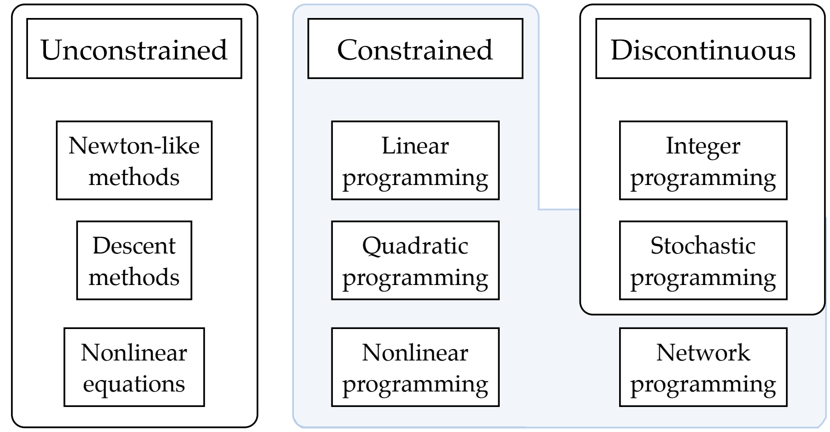
Constrained Optimization

Optimization Taxonomy



Quadratic Programming

- Find the minimum (maximum) value of a quadratic objective function subject to linear constraints
  - Lagrangian multipliers
  - Active set method

Equality Constraints

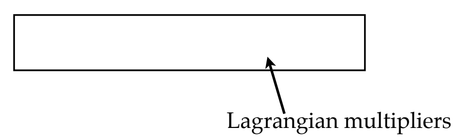
Suppose we want to minimize

$$F(\mathbf{x})$$

subject to $m \leq n$ equality constraints

$$\mathbf{C}(\mathbf{x})$$

The classical approach is to define the Lagrangian



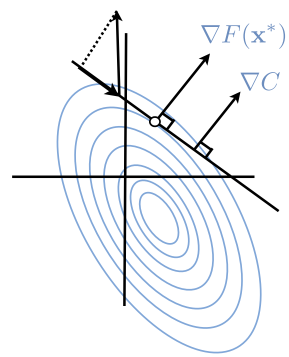
Equality constraints

Minimizing the Lagrangian is equivalent to solving

$$\nabla_x L(\mathbf{x}, \lambda) = 0$$
$$\nabla_\lambda L(\mathbf{x}, \lambda) = 0$$

$$\nabla_x L = \mathbf{g} - \mathbf{J}^T \lambda = \nabla F - \sum_{i=1}^{m} \lambda_i \nabla C_i$$
$$\nabla_\lambda L = -\mathbf{C}(\mathbf{x})$$

Equality constraints



At the optimal solution, the gradient of the objective function is the linear combination of the constraint gradients

The projection of the gradient of the objective function onto the constraint surface is zero at the optimal solution

Lagrangian-Newton

Apply the Newton method to find $(\mathbf{x}, \boldsymbol{\lambda})$ that minimizes the Lagrangian

$$L(\mathbf{x}^*, \lambda^*) = L(\mathbf{x}_0, \lambda_0) + \begin{bmatrix} \nabla_x L \\ \nabla_\lambda L \end{bmatrix}^T \mathbf{p} + \frac{1}{2}\mathbf{p} \begin{bmatrix} \nabla_{xx} L & \nabla_{x\lambda} L \\ \nabla_{\lambda x} L & \nabla_{\lambda\lambda} L \end{bmatrix} \mathbf{p}$$

$$\nabla_{xx} L = \mathbf{H} \qquad \nabla_{x\lambda} L = -\mathbf{J} \qquad \nabla_{\lambda x} L = -\mathbf{J}$$

$$\begin{bmatrix} \mathbf{H} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 - \mathbf{x}^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{C} \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) system

Inequality constraints

Suppose we want to minimize

$$F(\mathbf{x})$$

subject to inequality constraints

$$\mathbf{C}(\mathbf{x}) \geq \mathbf{0}$$

Collection of all points that satisfy the constraints is called the *feasible* region

Constraints can be partitioned into two sets: active set A and inactive set A'

Identify the active constraints $\tilde{\mathbf{C}}$ at each iteration and solve for

$$L(\mathbf{x}, \lambda) = F(\mathbf{x}) - \lambda^T \tilde{\mathbf{C}}(\mathbf{x})$$

Active Set Strategy

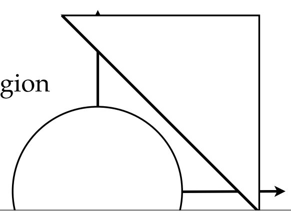
How do we identify an active set?

$$\lambda^* \geq 0 \quad \text{for} \quad i \in A$$

Example: minimize $F(\mathbf{x}) = x_1^2 + x_2^2$

subject to $C(\mathbf{x}) = 2 - x_1 - x_2 \geq 0$

feasible region



Quadratic Programming

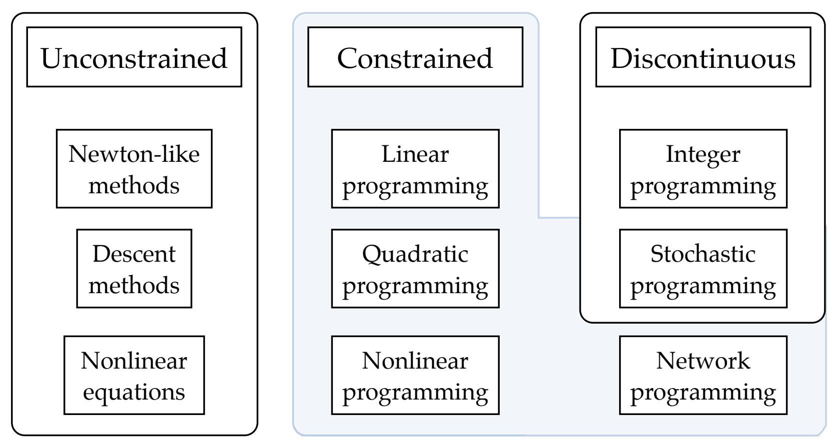
- A quadratic objective $F(\mathbf{x}) = \mathbf{g}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$
- Linear constraints $\mathbf{A}\mathbf{x} = \mathbf{a}$ and $\mathbf{B}\mathbf{x} \geq \mathbf{b}$

Assume that an estimate of the active set A^0 is given in addition to a feasible point $\mathbf{x}^0$

Quadratic Programming

- 1. Solve the KKT system with equality constraints and inequality constraints in the active set
- 2. Take the largest possible step size $\alpha \leq 1$ in the direction p that does not violate any inactive inequalities
- 3. If $\alpha < 1$, then add the limiting inequality to the active set A and return to step 1. Otherwise, take a full step and check λ
  - if all λ are positive, terminate the program
  - otherwise, delete the inequality with the most negative λ from A and return to step 1

Optimization Taxonomy



AN SQP ALGORITHM

- Sequential quadratic programming solves NLP by formulating a sequence of QP subproblems
- The solution $\mathbf{p}_{(k)}$ to each QP determines the search direction
- Each QP is formulated by
  - Quadratic approximation to the objective function
  - Linear approximation to the constraints

GLOBALIZATION STRATEGIES

- Merit functions

$$F(\mathbf{x}) + \frac{\sigma}{2}\mathbf{C}^T(\mathbf{x})\mathbf{C}(\mathbf{x})$$

- Line search methods
- Trust region methods

$$\frac{1}{2}\mathbf{p}^T\mathbf{p} \leq \delta^2$$

SUMMARY

- What is the physical meaning of Lagrangian multipliers?
- How does Lagrangian enforce "hard" constraints?
- How is a nonlinear problem solved using quadratic programming?