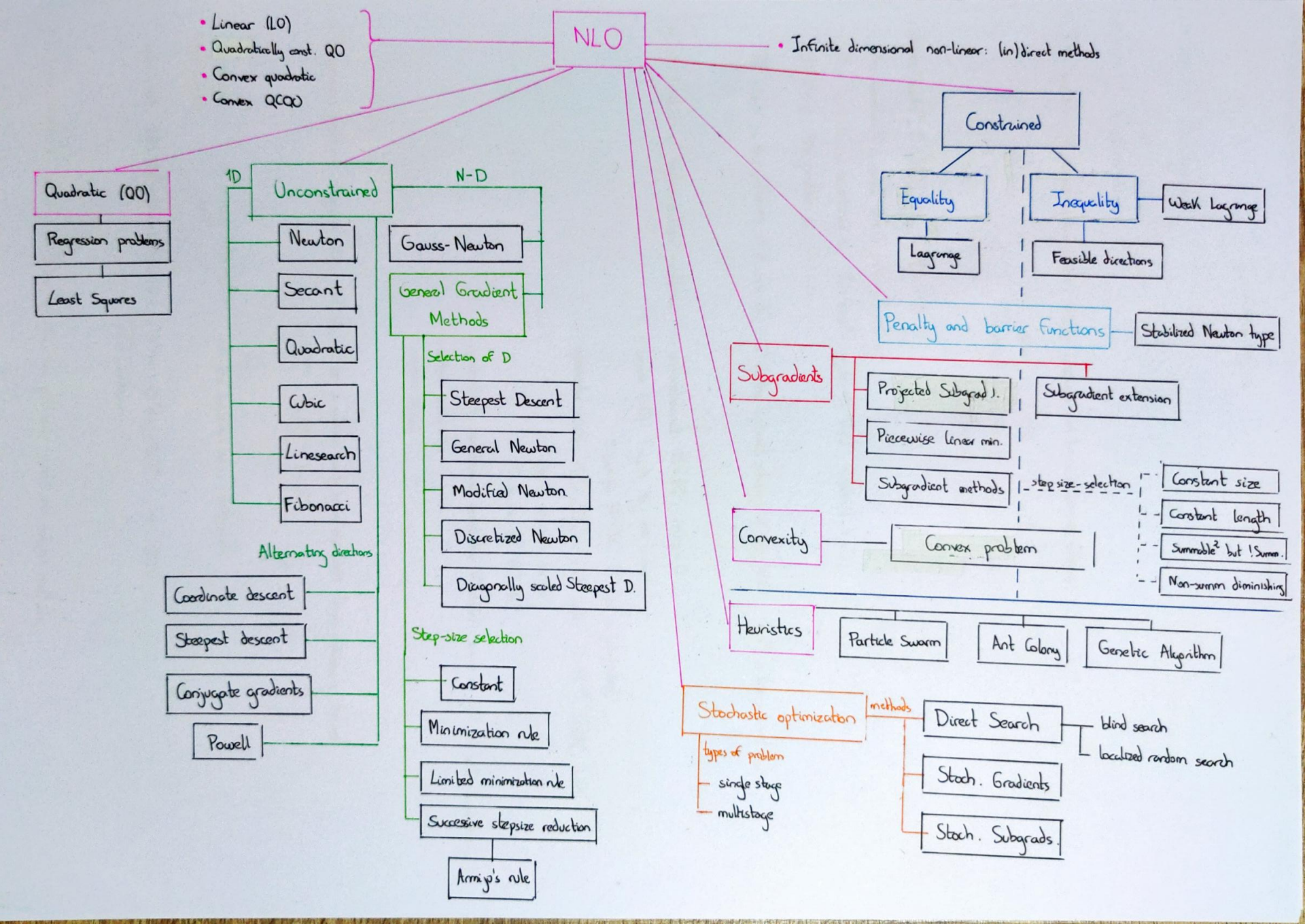
NLO
• Linear (LO)
• Quadratically const. QO
• Convex quadratic
• Convex QCQO
• Infinite dimensional non-linear: (in)direct methods
Quadratic (QO)
Regression problems
Least Squares
Unconstrained
1D
N-D
Newton
Secant
Quadratic
Cubic
Linesearch
Fibonacci
Gauss-Newton
General Gradient Methods
Selection of D
Steepest Descent
General Newton
Modified Newton
Discretized Newton
Diagonally scaled Steepest D.
Alternating directions
Coordinate descent
Steepest descent
Conjugate gradients
Powell
Step-size selection
Constant
Minimization rule
Limited minimization rule
Successive stepsize reduction
Armijo's rule
Constrained
Equality
Lagrange
Inequality
Weak Lagrange
Feasible directions
Penalty and barrier functions
Stabilized Newton type
Subgradients
Projected Subgrad.
Piecewise linear min.
Subgradient methods
Subgradient extension
step size selection
Constant size
Constant length
Summable² but ! Summ.
Non-summ diminishing
Convexity
Convex problem
Heuristics
Particle Swarm
Ant Colony
Genetic Algorithm
Stochastic optimization
methods
types of problem
single stage
multistage
Direct Search
blind search
localized random search
Stoch. Gradients
Stoch. Subgrads.

Optimization Final Exam Summary

Lecture 0. Background

Lecture 1. Optimization

- The general nonlinear optimization (NLO) problem can be written as follows:

$$\begin{array}{lll} \min & f(x) & x \in \mathcal{C} \\ \text{subject to} & g_i(x) = 0 & i \in \mathcal{I} = \{1, \dots, m\} \\ & h_j(x) \leq 0 & j \in \mathcal{J} = \{1, \dots, p\} \end{array}$$

where $x \in \mathbb{R}^n$, $\mathcal{C} \subset \mathbb{R}^n$, $f, g_i, h_j : \mathcal{C} \to \mathbb{R}$.

- The function f is called the objective function.
- The feasible set is defined by $\mathcal{F} = \{x \in \mathcal{C} : g_i(x) = 0 \ \forall i \in \mathcal{I}, \ h_j(x) \leq 0 \ \forall j \in \mathcal{J}\}$
  If $\mathcal{F} = \emptyset$, the problem is unfeasible.
- If $x^* \in \mathcal{F}$ is the infimum of f over $\mathcal{F}$, x^* is the optimal solution of the NLO. $f(x^*)$ is the optimal value.

- Classification of optimization problems:
  - unconstrained: $\mathcal{C} = \mathbb{R}^n$, $m = p = 0$
  - linear (LO): $f, g_i, h_j \ \forall i,j$ are linear
    $\mathcal{C}$ can be $\mathbb{R}^n$, $\mathbb{R}^n_+$, $\mathbb{R}^n_-$, or a polyhedral.
  - quadratic (QO): $f(x) = x^T Q x + c^T x + d$ with $Q \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$
    $g_i, h_j \ \forall i,j$ are linear.
    $\mathcal{C}$ can be $\mathbb{R}^n$, $\mathbb{R}^n_+$, $\mathbb{R}^n_-$
  - quadratically constrained quadratic (QCQO): same as QO but with quadratic g_i, h_j
  - convex quadratic
  - convex QCQO

- Regression problems (type of QO): Let $Ax = b$ be a system with more equations than unknowns. $(m > n)$
  $A \in \mathbb{R}^{m \times n}$ $x \in \mathbb{R}^n$ $b \in \mathbb{R}^m$

$\Longrightarrow$ least squares $\quad x^* = \min_{x \in \mathbb{R}^n} \|Ax - b\|$ with the Euclidean norm $\|x\| = \sqrt{x^T x}$

note that: $\|Ax-b\|^2 = (Ax-b)^T(Ax-b) = \underbrace{x^T A^T A x}_{\geq 0 \text{ if } \operatorname{rank}(A) > n} - 2b^T A x + \|b\|^2 \Rightarrow$ QO.

- Infinite dimensional general non-linear optimization problems: direct and indirect methods.

Lecture 2. Unconstrained and constrained optimization with equalities. Optimality conditions.

- **Sufficient conditions for extrema:** If c has a maximum of f: f increases in $(c-\delta, c)$, f decreases in $(c, c+\delta)$, $\delta > 0$ ⊛

- **Necessary condition for extrema:** $\nabla f(c) = 0$

⊛ → translated in Theorem: local minimum if $\nabla f(x^*) = 0$, $z^T \nabla^2 f(x^*) z > 0 \ \forall z \neq 0$

local maximum if $\nabla f(x^*) = 0$, $z^T \nabla^2 f(x^*) z < 0 \ \forall z \neq 0$

- **Convexity:** efficiency, every extrema is global, $\nabla f(x^*) = 0$ is sufficient (and necessary), many convex reformulations
  - A set $\mathcal{C}$ is convex if $\forall x_1, x_2 \in \mathcal{C} \ \forall \lambda \in [0,1] \quad \lambda x_1 + (1-\lambda) x_2 \in \mathcal{C}$
  - A function f is convex if $f(\lambda x_1 + (1-\lambda) x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ ($<$ for strictly convex.) $f: \mathcal{C}$ convex $\to \mathbb{R}$
    - ↳ at most one global minimum

  If f is convex, $-f$ is concave.

- **Optimization with equality constraints:** $f: C \to \mathbb{R}$, $\min/\max f(x)$ s.t. $g_i(x) = 0 \quad i = 1, \dots, m \quad m < n, \ C \subset \mathbb{R}^n$

1) Eliminate m variables through g_i (under the conditions of Implicit Function Theorem) → minimize for the rest: may be impossible

2) **Lagrange multipliers:** transform the problem into an unconstrained one:

$$L(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i g_i(x) \quad \to \text{Lagrange function}$$

- Necessary conditions:
  - $f: C \subset \mathbb{R}^n \to \mathbb{R}$, $g_i: C \subset \mathbb{R}^n \to \mathbb{R}$
  - x^* is local extrema of f in $N_\varepsilon(x^*)$
  - If $x \in N_\varepsilon(x^*)$, $g_i(x) = 0 \ \forall i$
  - g_i continuously derivable in $N_\varepsilon(x^*)$
  - The Jacobian matrix $\left(\frac{\partial g_i(x^*)}{\partial x_j}\right)_{i,j}$ has rank m

  } $\exists \lambda^* / \nabla L(x^*, \lambda^*) = 0$

- Sufficient conditions:
  - f, g_i, twice continuously differentiable real-valued functions in $\mathbb{R}^n$
  - $\exists x^*, \lambda^* / \nabla L(x^*, \lambda^*) = 0$
  - $\forall z \in \mathbb{R}^n, z \neq 0 \ / \ z^T \nabla g_i(x^*) = 0 \ \forall i$ it follows that $z^T \nabla_x^2 L(x^*, \lambda^*) z > 0$

  } x^* local minimum of f s.t. $g_i(x^*) = 0 \ \forall i$

Lecture 3. Methods for unconstrained optimization.

- n-order methods use up to n-order derivatives: includings gradients, hessians, etc.

- **Newton's method:** guess first point x_0 s.t. $f'(x_0)=0$

  iterate as $$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)} = \phi(x^k)$$

  - Necessary conditions: • ϕ real valued and continuous in $\phi:[a,b]\to T\subset\mathbb{R}$
    • ϕ contractive: $\exists q\in(0,1) \mid \forall x^1, x^2\in[a,b] \quad |\phi(x^1)-\phi(x^2)|\le q|x^1-x^2|$

    $\Big\}\ \forall x^0\in[a,b]\ \{x^k\}\longrightarrow x^*$ (unique)

  - Sufficient condition: • $|\phi'(x)|<1 \ \forall x\in[a,b] \Rightarrow \phi$ contraction

- **Secant method:** $$x^{k+1} = x^k - \frac{f'(x^k)(x^k - x^{k-1})}{f'(x^k) - f'(x^{k-1})}$$

- **Quadratic Method:** (polynomial approximation) $\phi(x) = a+bx+cx^2 \approx f(x)$

  ↓ Find by imposing $\phi(x_i)=f(x_i)$ for $i=1,2,3 \mid x_1<x_2<x_3$

  if $c<0$ → unusable

  if $c>0$ → first approximation: $\tilde{x} = -\frac{b}{2c}$ if $f(x_1)>f(x_2)$, $f(x_3)>f(x_2) \Rightarrow$ local minimum between x_1 and x_3.

  consider $x_1, x_2, x_3, \tilde{x}$: choose the minimum (new x_2) and the two adjacent (new $x_{1,3}$)

  repeat.

- **Cubic Method:**
  - $f(x)\approx\phi(x) := a+bx+cx^2+dx^3$
  - first derivatives must be evaluable

  $\longrightarrow$ solve $f(x_1)=\phi(x_1)$; $f(x_2)=\phi(x_2)$ (↱ $f'(x_1)<0$ and $\begin{cases} f'(x_2)\ge 0 \\ \text{or} \\ f(x_2)>f(x_1) \end{cases}$)

  $f'(x_1)=\phi'(x_1)$; $f'(x_2)=\phi'(x_2)$

- **Linesearch method:** • f must be unimodal in $[a,b]$ (only one minimum in $[a,b]$)

  1) Evaluate at 2 points x_1, x_2
  2) Depending on $f(x_1)$, $f(x_2)$ values, choose, the interval where the search must continue.
  3) Repeat

- **Fibonacci method:** minimizes the length of possible interval after N (↙ needs this number though) evaluations among unimodal minimization methods

$$x_1^k = l_k + \frac{F_{N-k}}{F_{N+2-k}}(r_k - l_k), \quad x_2^k = l_k + \frac{F_{N-k+1}}{F_{N+2-k}}(r_k - l_k) \quad \text{if } x_*^k \in [l_k, r_k]$$

- Golden section method: approximation of Fibonacci without specifying number of iterations. ⊕

$$x_1^{kG} = l_k + \frac{\tau-1}{\tau}(r_k - l_k) \qquad x_2^{kG} = l_k + \frac{1}{\tau}(r_k - l_k)$$

where τ is the golden ratio.

↓ n-dimensional unconstrained methods.

- General Gradient Methods: $x^{k+1} = x^k + \alpha^k d^k$ (step, direction)

  - Generally, the direction is defined as: $d^k = -D^k \nabla f(x^k)$ (D^k: pos. def matrix)

  - Methods defined by D selection:

    * Steepest descent: $D^k = Id \ \forall k$. slow convergence ⊖

    * General Newton's method: $D^k = (\nabla^2 f(x^k))^{-1}$ Fast convergence ⊕ second order ⊖

    * Modified Newton's method: $D^k = (\nabla^2 f(x^0))^{-1}$ avoids Hessian computed at each step ⊕

    * Discretized Newton's method: $D^k \approx (\nabla^2 f(x^k))^{-1}$ ⊖ approximation done through values of f' ⊕

    * Diagonally scaled steepest descent $D^k = \text{diag}(d_1^k, \dots, d_n^k)$ with $d^k \approx \left(\frac{\partial^2 f(x^k)}{\partial x_i^2}\right)^{-1}$

  - Stepsize selection:

    * Constant: $\alpha^k = s \ \forall k$

    * Minimization rule: α^k / $\min_{\alpha \geq 0} f(x^k + \alpha^k d^k)$

    * Limited min. rule: same as above but imposing $\alpha^k < s$

    * Successive stepsize reduction. reduce until $f(x^k + sd^k) < f(x^k)$ is fulfilled

    (convergence if $\{d^k\}$ is gradient related to $\{x^k\}$)

    ↓ Armijo's rule

    cost improvement may not guarantee convergence ⊖

vs. Newton: { no second derivatives ⊕ ; slower convergence ⊖ }

- Gauss-Newton Method: $\min F(x) = \frac{1}{2}\sum_{i=1}^{m} g_i^2(x)$ → uses $g(x) \approx g(x^k) + \nabla g(x^k)^T (x - x^k)$

$$x^{k+1} = x^k - \alpha^k \underbrace{\left[\nabla g(x^k) \nabla g(x^k)^T\right]^{-1}}_{D^k} \nabla F(x^k)$$

Lecture 4: Alternating directions methods for unconstrained optimization.

base: 1-D min. always change direction.

- **Coordinate descent method:** use n orthogonal vectors $e_1, \dots, e_n$ as d_i one per step.

  $\min f(x + \alpha e_i)$

  

  slow convergence

- **Steepest descent:** $d^{k+1} = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$ → always changes! (orthogonally to previous)

  no right angle turns

  ⊖

- **Conjugate directions** guarantee perservation of the minimization achieved in previous steps ⊕
  - $x, y \in \mathbb{R}^n$ are conjugate directions w.r.t. nxn symmetric pos. def. matrix A if $x^T A y = 0$

    any $\langle v_1, \dots, v_k \rangle$ l.i vectors can be reconstructed to $\langle d_1, \dots, d_k \rangle$ conjugate.
  - **Theorem:** the minimum of $f(x) = a + b^T x + \frac{1}{2} x^T A x$ can be found by searching only once per conjugate direction. ⊕

- **Conjugate gradient methods:** $x^{k+1} = x^k + \alpha_{k+1} z^{k+1}$
  - quadratic functions: $f(x) = a + b^T x + \frac{1}{2} x^T A x \longrightarrow \alpha_k^* = -\frac{(z^k)^T \nabla f(x^{k-1})}{(z^k)^T A z^k}$
  - in general (choice of directions) → $z_1 = -\nabla f(x^0)$ … $z^{k+1} = -\nabla f(x^k) + \sum_{j=1}^{k} \beta_{kj} z^j$

    $\beta_{kj} = 0 \quad k \neq j$ $\qquad \beta_{kk} = \frac{(\nabla f(x^k))^T \nabla f(x^k)}{(\nabla f(x^{k-1}))^T \nabla f(x^{k-1})}$
  - non quadratic: α_k^* computed with any 1-D min // long computations of β and $\nabla^2 f$

    ⊖

- **Powell's Method:** no derivatives ⊕ ; only needs continuity ⊕
  1) Each stage: n+1 1D line searches
  2) First n → n l.i. directions
  3) Last direction is the one connecting best point (from n searches) to stage starting point.
  4) The (n+1)th direction substitutes one of n first directions and repeat.

  ↳ Finite steps on quad. functions ⊕

  problems with linear dependent directions ⊖

Lecture 5. Constrained optimization

↓ equality constrained: $\min f(x)$ s.t. $h_j(x)=0$

- Necessary conditions:
  - $x^* \in X$ is a solution of the constrained problem
  - at $x=x^*$ $J = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \end{pmatrix}$ has rank m.

  $\Rightarrow \exists \lambda^* \mid \nabla L(x^*, \lambda^*) = 0$

↓ general + s.t $g_i(x) \geq 0$

- Necessary conditions:
  - $x^* \in X$ is solution
  - $(Z^1(x^*))' - (S(X, x^*))'$

  Kuhn-Tucker: $\exists \lambda^* \in \mathbb{R}^p$, $\mu^* \in \mathbb{R}^m$ | $\nabla f(x^*) - \sum_i^p \lambda_i^* \nabla g_i(x^*) - \sum_j^m \mu_j^* \nabla h_j(x^*) = 0$, $\lambda_i^* g_i(x^*) = 0$, $\lambda^* \geq 0$

↓ equality

- Sufficient:
  - $\nabla L(x^*, \lambda^*) = 0$ ≡ (or KKT conditions for general)
  - $\forall z \in \mathbb{R}^n \setminus \{0\}$ s.t $z^T \nabla h_i(x^*) = 0 \Rightarrow z^T \nabla^2_{xx} L(x^*, \lambda^*) z > 0$ ⟹ x^* strict local minimum

  $z \in Z^{-1}(x^*)$; $L(x^*, \lambda^*, \mu^*)$ in gen.

↓ finding equality constrained extrema.

- Lagrange's Method: seen. Feasible variations are introduced.

↓ Finding inequality constrained extrema.

- Feasible directions are those that point towards a minimum's neighborhood from the current point.

  ↳ space: $Z^1(x^*) = \{z \mid z^T \nabla g_i(x^*) \geq 0;\ z^T \nabla h_j(x^*) = 0\ \forall j\}$

  ↓ $\forall i \mid g_i(x^*) = 0$

- Weaker Lagrangian as weaker optimality condition: $\tilde{L}(x, \lambda, \mu) = \lambda_0 f(x) - \sum_{i=1}^{p} \lambda_i g_i(x) - \sum_{j=1}^{m} \mu_j h_j(x)$

  ↓ additional

  is ignored in some theorems.

Lecture 6. Penalty and barrier function methods for constrained optimization.

- **Penalty function methods:** transforms constrained into unconstrained.

infinite penalty: $o(x) = \begin{cases} 0 & \text{if } x \in X \\ +\infty & \text{if } x \notin X \end{cases}$ → feasible set

$\min_{x \in \mathbb{R}^n} F(x) = \min_{x \in \mathbb{R}^n} (f(x) + o(x))$ ⟹ x^* minimizes F in $\mathbb{R}^n$ ⟺ x^* minimizes f in X

↓ discontinuity in boundaries of X (⊖)

↓ infinite values outside of X (⊖)

$o(x) = \begin{cases} 0 & \text{for } x < 0 \\ \lambda x^3 & \text{for } x \geq 0 \end{cases}$

Use unconstrained algorithm (Golden section) on $\tilde{F} := F$ → it will get reasonably close. (try increasing λ) (⊖: low convergence, large gradients.)

- penalty function: (parameter λ, variable t)
  1) $\phi(\lambda, t)$ continuous
  2) $\phi(\lambda, t) \geq 0 \quad \forall \lambda, t$
  3) $\phi(\lambda, t) = 0 \quad \forall t \leq 0$, and strictly increasing for $\lambda \geq 0$ and $t > 0$

  ⟹ $\phi(\lambda, t) = \begin{cases} 0 & \text{for } t < 0 \\ \lambda t^n & \text{for } t \geq 0 \end{cases}$

  Desirable one/two continuous derivatives in t.

we can modify objective function: $\tilde{F}(x) = f(x) + \sum_i^p \phi(\alpha_i, g_i(x)) + \sum_j^m [\phi(\beta_j, h_j(x)) + \phi(\beta_j, -h_j(x))]$

- ↑ minimize with no constraints.
- α_i, β_j: strength of constraint enforcement
- penalize if $h_j \neq 0$

⊕ no initial feasible point choice
most constraints in real world are soft
constrained → unconstrained

⊖ no exact solutions
sometimes unapplicable as function may be undefined outta feasi. set
increase λ ⟹ ill condition

- **Barrier function methods** type of penalty function only for inequality constrained that always mantains feasible iterates.

$$\min f(x) \quad \text{s.t} \quad g_i(x) \geq 0$$

logarithmic: $\phi(x) = -\sum_{i=1}^p \log(g_i(x))$ inverse: $\phi(x) = \sum_{i=1}^p \frac{1}{g_i(x)}$ → approaches to penalty functions.

1) $\tilde{F}_\mu(x) = f(x) + \mu\phi(x)$ (barrier param. μ) → start $\mu > 0$ and feasible point x_0.
2) minimize $\tilde{F}$
3) decrease μ and re-optimize using final iterate as first one for new μ
4) Repeat.

⊕ given barrier functions are convex

⊖ potential difficulties
increasingly difficulty as μ decreases
very ill conditioned

- Penalty functions for equally constrained problems: $\min f(x)$ s.t $h(x)=0$

$\phi(x) = \begin{cases} 0 & \text{if } x \text{ is feasible} \\ >0 & \text{otherwise} \end{cases}$ $\Rightarrow \phi(x) = \frac{1}{2} h(x)^T h(x)$ quadratic-loss function.

define $\tilde{f}_{p_k}(x) = f(x) + p\phi(x)$ and minimize unconstrained problems for an increasing sequence $\{p_k\}$
↓ increases → approaches to 'ided' penalty

⊕ convergence under mild conditions
sequence of penalty function minimizers defines continous trajectory

⊖ p increases, more ill-conditioned.

- Penalty functions for inequality constrained problems $\min f(x)$ s.t. $g(x) \geq 0$

$\phi(x) \begin{cases} =0 & \text{if } x \text{ is feasible} \\ >0 & \text{otherwise} \end{cases}$ $\Rightarrow \phi(x) = \frac{1}{2} \sum_{i=1}^{p} [\min(g_i(x), 0)]^2$

- Stabilized penalty and barrier method: Newton-type direction. numerically stable.

Lecture 7. Heuristic optimization methods

For large dimension problems. at reasonable computational cost (⊕)

- Determines an optimal solution by trying to improve a candidate solution.

routine:
- initialize: fix population size of M
  define fitness/objective function f and constraints g,h
  random position of M members in search space.
  define stopping criteria and max_iter
- while iters:
  increase population
  evaluate fitness
  choose fittest, discard weaker ones.
  stop? iters?

member of population with highest fitness is global optimal solution.

• Particle Swarm: population of particles → flow through space defining trajectories driven by own and best neighbors performances. → cooperative.

- initialization: random particle initial positions
  initialize fitness, best positions and global best.
- routine: update particles' velocity, position, fitness
  update bests and global best
  stopping criteria?

④ w = inertia. w<<1 no info from the past is used
w ≥ 1 swarm diverges } generally [0,3 0,9]

there are constants that measure the influence of personal and global bests.
R_1, R_2 matrices of random numbers that give a stochastic influence on both cognitive and social components.
trajectories are pseudo-random.

• Ant colony: For problems which can be reduced to finding good paths in graphs.
ants follow pheromone trails other ants leave to find shorter paths choosing probabilities marked by strong pheromone concentrations
the ants paths converge to the same one.

$G = (D, P)$ { D = {nest, food}
P = paths → imagine 2 (initially each path will be chosen by 50%) → prob. will change

↑ Type of Evolution Programs (GA, Evolution Strategies, Evolutionary Programming, Scatter Search)

• Genetic algorithm: stochastic breeding methods that mimic biological evolution
select individuals according to fitness + breeding.

- Initialize: initial population M, define fitness function f
  encode population as chromosomes (bit string)
  compute fitness of entire population
  define stopping criteria and max_iter
- While: selection of parents to reproduce
  crossover to produce offspring } genetic operators / high order \ small change
  mutation
  Fitness → select next gen members
  stop?

Lecture 8: Convexity

- Intersection of convex sets is convex ⇒ Convex hull $C(A)$: intersection of all convex sets containing a set A. (Closure convex) ↷ polyhedra if A is finite
- $X+Y=\{x+y \mid x\in X,\ y\in Y\}$. $\lambda X=\{\lambda x \mid x\in X\}$ are also convex if X and Y are.
- C convex, $v\in C$ is an extremal point if it can't be written as a convex combination ($v=\sum_{i=1}^{m} a_i v_i$ with $\sum a_i=1,\ a_i>0$)
  circunference of a circle, vertices of a triangle.

  If C convex, bounded, closed and with finite extremal points, $\forall v\in C$ can be written as a convex combination of the extremal points.
- $S\subset\mathbb{R}^n$ is a cone if $v\in S \Rightarrow \lambda v\in S,\ \forall\lambda\geq 0$  the origin is always in a cone. not all cones are convex.
- Simplex: n-dim convex polyhedra with $n+1$ vertices (point, segment, triangle, tetrahedron): the faces are lower dim simplex.
  - unitary if $x_i\geq 0$ $\sum_{i=1}^{n} x_i\leq 1$ tetrahedron $(0,0,0)$ $(1,0,0)$ $(0,1,0)$ $(0,0,1)$

- All this allows us to transform a set of inequalities into a set of equalities using slack variables
  - Linear programming: given a convex set defined by a linear set of constraints determine in which subset (can be a point) a certain linear point has its maximum or minimum.
  - the epigraph of $f_1(x)=\begin{cases} f(x) & \text{if } x\in D \\ \infty & \text{if } x\notin D \end{cases}$ is the same as f's ⇒ construction of convex functions defined in all $\mathbb{R}^n$
  - proper convex functions: $f(x)>-\infty\ \forall x$, $f(x)\neq\infty\ \forall x$
- Discontinities can only occur at some boundary points ⇒ avoidable through the closure operations.
- Differential properties: convexity of a function in $\mathbb{R}^n$ is equivalent to its convexity in any line restriction in $\mathbb{R}^n$
  ⇒ study the results in $\mathbb{R}$: it's enough and simpler.
  - Also, f diff at open interval $D\subset\mathbb{R}$. f' non-decreasing in $D \Leftrightarrow f$ convex in D.
- Let f be proper convex: all minima are global (on $\mathbb{R}^n$ or X)
  ↳ on $\mathbb{R}^n$ or convex set $X\subset\mathbb{R}^n$
  - it must be unique.
  - $\nabla f(x^*)=0 \Leftrightarrow x^*$ minimum
- Convex problem: ↷ proper convex

$$\begin{aligned} \min\ & f(x) \\ \text{s.t}\ & g_i(x)\geq 0 \\ & h_j(x)=\sum_{k=1}^{n} a_{jk}x_k-b_j=0 \end{aligned} \Big\} \text{ convex feasible set.}$$

  - Sufficient condition: general constraints Karuhn-Kush-Tucker conditions.

Lecture 9: Subgradient methods for convex problems

remember Gradient methods. (background)

- **Subgradient:** at a point $x \in \mathbb{R}^n$ of a conver function is ony vector $g \in \mathbb{R}^n$ such that $\forall y \in \mathbb{R}^n$

$$f(y) \geq f(x) + g^T(y-x)$$

can be 0, 1 or more vectors satisfying this. $\Rightarrow$ set of subgradients: $\partial f(x)$ subdifferential

- $\partial f(x)$ is a closed set.
- $\#\partial f(x) = 1 \Leftrightarrow f$ is differentiable at x
- x^* is a minimizer of $f \Leftrightarrow 0 \in \partial f(x^*) \neq \emptyset$
- $\partial(\alpha f)(x) = \alpha \partial f(x) \quad \forall \alpha \geq 0$
- $f = f_1 + \dots + f_m \Rightarrow \partial f(x) = \partial f_1(x) + \dots + \partial f_m(x)$

- **Subgradient methods:** for non-differentiable convex functions when high accuracy is not needed (10%)

⊖ slow
no good stopping criterion

⊕ simple algorithms
used to decompose bigger problems
lower memory requirements.

Let $f: \mathbb{R}^n \to \mathbb{R}$ be convex: $x_{k+1} = x_k - \alpha_k g_k$

↳ ony subgradient of f at x_k

it is not a descent method $\Rightarrow$ Keep track of best result

- step size rules:
  - * Constant size
  - * Constant length = $\alpha_k = \frac{\|x_{k+1} - x_k\|_2}{\|g_k\|_2}$
  - * Square summable but not summable: $\sum_{k=1}^{\infty} \alpha_k^2 < \infty \quad \sum_{k=1}^{\infty} \alpha_k = \infty \quad \alpha_k = \frac{a}{b+k} \quad a, b > 0$
  - * Nonsummable diminishing: $\lim_{k \to \infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty \quad \alpha_k = \frac{a}{\sqrt{k}}$

- **Projected subgradient method** (extension): $\min f(x)$ s.t $x \in C$, C convex : $x_{k+1} = P(x_k - \alpha_k g_k)$ ↓ euclidean projection in C

- **Piecewise linear minimization:** $\min f(x) = \max_{i=1,\dots,m} (a_i^T x + b_i)$ → Find j / $j = \operatorname{argmax} (a_i^T x + b_i)$: $x_{k+1} = x_k - \alpha_k a_j$

- **Polyak's step choice** (for when optimal value f^* is known): $\alpha_k = \frac{f(x_k) - f^*}{\|g_k\|_2^2}$ → even with this: slow

- with estimated $f^* = f^i_{best} - \gamma_i$ $\quad \alpha_i = \frac{f(x_i) - f^i_{best} - \gamma_i}{\|g_i\|_2^2}$ → estimate of how suboptimal our current point is.

If $\sum_{i=1}^{\infty} \gamma_i = \infty$, $f^i_{best} \to f^*$

- Finding a point in $C = C_1 \cap \dots \cap C_m$, all non-empty convex sets.

$$\min F(x) = \max\{dist(x, C_1), \dots, dist(x, C_m)\}$$

in this case, $g = \nabla dist(x, C_j) = \dfrac{x - P_{C_j}(x)}{\|x - P_{C_j}(x)\|}$ — P Euclidean projection onto C_j

using subgradient method with Polyak's estimate: $x_{k+1} = P_{C_j}(x_k)$ → project current point onto farthest set.

- Subgradient method extension for inequality constraints.

$$\min f_0(x)$$
$$\text{s.t. } f_i(x) \leq 0 \quad i = 1, \dots, m$$

where f_i convex

$$x_{k+1} = x_k - \alpha_k g_k$$

$$g_k \in \begin{cases} \partial f_0(x_k) & \text{if } f_i(x_k) \leq 0 \ \forall i = 1, \dots, m \\ \partial f_j(x_k) & \text{if } f_j(x_k) > 0 \end{cases}$$

current point → Feasible → objective subgradient (quality as constrained notion)

→ non-feasible → try to fix violated constraint

also needs to keep track of so-far best.

Lecture 10. Stochastic optimization methods.

- Inherent uncertainty: random errors, in measurements ① — Monte Carlo simulations. search direction random choice

① $F(x) = f(x) + \varepsilon(x)$

- Stochastic optimization methods use random variables in the shape of objective functions and/or constraints.

- Direct random search methods: with noise-free measurements of $f(x)$

⊕ easy code
no need for gradients, etc.
efficient
applicable to non-trivial objective functions.

- Blind Search: 1) random or deterministic (probabilistic distribution) $x^0 \in X$, compute $f(x^0)$, $k = 0$

through prob. dist ← 2) compute $x^{k+1}_{new} \in X$. If $f(x^{k+1}_{new}) < f(x^k)$, $x^{k+1} = x^{k+1}_{new}$. Otherwise $x^{k+1} = x^k$

3) max-iters?

- Localized random search: ⊕ global convergence.

  Change in step 2: generate independent random vector d^k until $x_{new}^{k+1} = x^k + d^k \in X$.

• Stochastic optimization problems: need to know statistical feat. of random params + order in which info enters and is used.

- Single stage: dynamics of entering initial information does not play a role.
  try to find single optimal decision with deterministic optimization methods.

  $\mathcal{X}$ feasible decisions, Find $x \in \mathcal{X}$ that minimizes $F(x, \xi)$ ← random information

  $$\min \mathbb{E}[F(x, \xi)]$$

  Sample average approximation (SAA) to solve $\xi^* = \min_{x \in X} \{f(x) = \mathbb{E}[F(x, \xi)]\}$

  1) Monte Carlo sampling because evaluation may not be possible

  $$\mathbb{E}[F(x, \xi)] \approx \frac{1}{n} \sum_{i=1}^{n} F(x, \tilde{\xi}_i) = F_n(x)$$

  2) Search: $\xi_n^* = \min_{x \in X} \{F_n(x)\}$

- Multistage: find an optimal sequence of decisions x_t, $t = 0, \dots, T$ (chess game)

  (2-stage)

  $$\min c^T x$$
  $$\text{s.t. } Ax = b$$
  $$\tilde{T}x = \tilde{\xi}$$

  → ill-conditioned

  ↓ stochastic variables based on probability

  Generical mathematical formulation: $\min c^T x + Q(x)$ s.t. $Ax = b$

  with $Q(x) = \mathbb{E}_{\tilde{\xi}}[\min \{\tilde{q}^T y \mid W y = \tilde{T}x - \tilde{\xi}\}]$

  ↓ 1st stage decision vector, ↘ 2nd

  "x represents the here-and-now" decision before upcoming random data $\tilde{\xi}$ comes.

  from 2-stage we can define n-stage iteratively: no good solution for all problems.

- Stochastic gradient method $f(x) = \frac{1}{N} \sum_{i}^{N} F_i(x)$

  $$x_{k+1} = x_k - a_k \nabla f_i(x_k) = x_k - a_k \frac{1}{n} \sum_{j_k=1}^{n} \nabla f_{j_k}(x_k)$$

  vs. deterministic: path is not straightforward

  gradient estimation from samples (unbiased)

- Stochastic subgradient methods: $x^{k+1} = x^k - a_k \tilde{g}^k$ with $\tilde{g} = g + v$, $g \in \partial f(x)$ noisy subgradient (v zero-mean)