ANO

Applied Numerical Optimization

Ng Huu Duc

L1, +) Definitions

- Variables : satisfy constraints, while [maximize / minimize] cost (compromise)
- Objective function
- A methematical model
- Additional Restrictions ≈ equality / inequality ‖ constraints

+) Application:
- Business
- Design
- Operational
- Model Identification

‖ Decisions

+) Classification:

- Linearity of objective function & constraints

- Discrete and/or continuous variables

- Time Dependence

- Stochastic or deterministic models & variables

- Single vs Multi objective, single level vs multi level

+) Common Terminology:

- Optimization problem: mathematical formulation (objective func(s), constraints, ..)
- An Algorithm: procedure ⇒ solve problem
- A Solver: software

+) Steps: System Analysis ⇒ $\left.\begin{array}{l}\text{Objective func(s)}\\ \text{Constraint(s)}\end{array}\right\} \Rightarrow$ Mathematical model + Variables

Classify → Select suitable algorithm, solver

Verify, try to understand

+) Issues / Challenges

⊛ Nonlinear Optimization Problem (Nonlinear Program - NLP)

$x = [x_1, x_2, .., x_n]^T \in \mathbb{D} \subseteq \mathbb{R}^n$ a vector

$\mathbb{D}$ host set

$f : \mathbb{D} \to \mathbb{R}$ objective function

$c_i : \mathbb{D} \to \mathbb{R}$ constraint functions $\forall i \in E \cup I$

- General formulation:

$$\min_{x \in \mathbb{D}} f(x)$$

$$\text{s.t.} \quad c_i(x) = 0, \ i \in E \quad \text{equality}$$
$$c_j(x) \leq 0, \ j \in I \quad \text{inequality}$$

constraints → algebraic / differential / probabilistic

- The feasible set:

$$\Omega = \{ x \in \mathbb{D} \mid c_i(x) \leq 0 \ \forall i \in I, \ c_i(x) = 0 \ \forall i \in E \}$$

⇒ Equivalent formulation $\min_{x \in \Omega} f(x)$

+) **Optimal solution:** – Solution Points

- x^* is **local solution** if $x^* \in \Omega$ and $\exists$ a neighborhood $N(x^*)$ of x^*
  $f(x^*) \leq f(x) \;\forall x \in N(x^*) \cap \Omega$
- x^* is **strict local solution** if $x^* \in \Omega$, $\exists\, N(x^*)$:
  $f(x^*) < f(x) \;\forall x \in N(x^*) \cap \Omega,\; x \neq x^*$
- x^* is **global solution** if $x^* \in \Omega$ and $f(x^*) \leq f(x) \;\forall x \in \Omega$
- x^* exists when $\begin{cases} \mathbb{D} \text{ is compact } (= \text{closed} + \text{bounded}) \\ f \text{ is continuous} \end{cases}$

L2) +) **Directional Derivative:**

Let $f: D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$, $x \in \mathbb{D}$ and $p \in \mathbb{R}^n$ with $\|p\| = 1$

f is differentiable at point $x = x_a$ in the direction p

if the limit $D(f,p)\big|_{x=x_a} = \lim\limits_{\varepsilon \to 0} \dfrac{f(x_a + \varepsilon p) - f(x_a)}{\varepsilon} =: \nabla_p f(x_a)$

exists and is finite

$D(f,p)$: the directional derivative of f in direction p.

+) **Gradient** of f at point x:

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}\Big|_x \;\; \cdots \;\; \frac{\partial f}{\partial x_n}\Big|_x \right]^T$$

- If x is time-dependent: $x(t)$

$$\Rightarrow \frac{df}{dt}\Big|_{x(t)} = \nabla f(x)^T \frac{dx}{dt}\Big|_t = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}\Big|_{x(t)} \cdot \frac{\partial x_i}{\partial t}\Big|_t$$

+) **Hessian matrix:** The 2nd derivative of a scalar, twice continuously differentiable function f is the symmetric Hessian matrix $H(x)$:

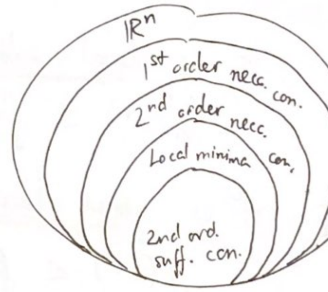
$$H(x) = \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}\Big|_x & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}\Big|_x \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}\Big|_x & \cdots & \frac{\partial^2 f}{\partial x_n^2}\Big|_x \end{bmatrix}$$

Conditions

+) <u>1st-order necessary conditions:</u>

Let f be continuously differentiable, let $x^* \in \mathbb{R}^n$ be a local minimizer of f

<u>then</u> $\nabla f(x^*) = 0$



- Stationary points: $\nabla f(x) = 0$
- Saddle points: are stationary points but not minimum / maximum

+) <u>2nd-order necessary conditions:</u>

- Let f be twice continuously differentiable
- $x^* \in \mathbb{R}^n$ be a local minimizer of f

<u>then</u>

1. $\nabla f(x^*) = 0$
2. $\nabla^2 f(x^*)$ is positive semidefinite

+) <u>Sufficient Optimality conditions:</u>

Let f be twice continuously differentiable <u>if</u>

x^* $\mathbb{R}^n$ ~~is local minimizer~~

1. $\nabla f(x^*) = 0$
2. $\nabla^2 f(x^*)$ is positive definite

<u>Then</u> x^* is a strict local minimizer of f

sufficient but <u>not necessary</u>

+) <u>Unconstrained optimization problem:</u> feasible set $\Omega = \mathbb{R}^n \Rightarrow \min_{x \in \mathbb{R}^n} f(x)$

- <u>Convex Set:</u>

Set $\Omega \subseteq \mathbb{R}^n$ is convex, if $\forall\, x_1, x_2 \in \Omega$ and $\forall\, \alpha \in [0,1]$

then $\alpha x_1 + (1-\alpha) x_2 \in \Omega$

Convex or non-convex (No concave sets)

+) Convexity of a Function:

$\forall x_1, x_2 \in \mathbb{D}, \forall \alpha \in [0,1]$

- f is convex on $\mathbb{D}$: $f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$
- ___ strictly convex ___ : ______ $<$ ______
- f is (strictly) concave on $\mathbb{D}$, if $(-f)$ is (strictly) convex
- smooth convex $\equiv$ convex + continuous 1^{st} & 2^{nd} derivatives

+) Matrix:

- A symmetrix matrix $A_{(n\times n)}$ is positive definite if $p^T A p > 0 \ \forall p \in \mathbb{R}^n, p \neq 0$

  semi ______ $\geq$

- Theorem: A is positive definite if all eigenvalues $\lambda_k > 0 \ \forall k$

  semi ______ $\lambda_k \geq 0$

- Convexity under differentiability:
  - f is convex, iff $H(x)$ is positive semi-definite $\forall x \in \mathbb{D}$
  - if $H(x)$ is positive definite $\forall x \in \mathbb{D} \Rightarrow f$ is strictly convex

⊗ Optimization proplem $\min_{x \in \Omega} f(x)$ is convex if $\begin{cases} f \text{ is convex} \\ \Omega \text{ is convex set} \end{cases}$

- Feasible set Ω is convex if $\begin{cases} \mathbb{D} \text{ is convex} \\ c_i \ \forall i \in \mathcal{I} \text{ are convex on } \mathbb{D} \\ c_i \ \forall i \in \mathcal{E} \text{ are linear} \end{cases}$

If f is smooth & convex $\Rightarrow \nabla^2 f(x) \geq 0 \ \forall x$

$\Rightarrow$ For convex opt. problem, 1st-order opt. condition is necessary and sufficient

ANO

Unconstrained Problem

L3)

Iterative Descent (Direct methods)	Optimality conditions (Indirect methods)
$\exists \bar{k} \geq 0 : f(x^{(k+1)}) < f(x^{(k)})$ $\forall k \geq \bar{k}$	$\nabla f(x) = 0 \Rightarrow$ nonlinear system of equations
$\lim_{k \to \infty} x^{(k)} = x^* \in \mathbb{R}^n$	solve anatically or numerically (complex)

+) Rate of Convergence:

- Linear: $\exists\ C \in (0,1)$, such that for sufficiently great k:
$$\|x^{(k+1)} - x^*\| \leq C \|x^{(k)} - x^*\|$$
- Order p (often $p=2$): $\exists\ M = \text{const}$ such that
$$\|x^{(k+1)} - x^*\| \leq M \|x^{(k)} - x^*\|^p$$
- Super linear: if there exists a sequence c_k converging to 0 $\lim_{k\to\infty} c_k = 0$ such that: $\|x^{(k+1)} - x^*\| \leq c_k \|x^{(k)} - x^*\|$

Line - Search Method:

$\nabla f(x^{(k)})^T p < 0$, p as descent direction, α as step length

$\Rightarrow$ Set $x^{(k+1)} = x^{(k)} + \alpha_k \cdot p^{(k)}$

+) Step length α_k calculation:

- Define descent direction ~~pk~~ $p^{(k)}$
- $\phi(\alpha) = f(x^{(k)} + \alpha . p^{(k)})$
- Choose α : $\min_{\alpha > 0} \phi(\alpha)$

+) Armijo Condition: often ask this in exam, not Wolfe

Let f be continuously differentiable, $p^{(k)}$ .. & $c_1 \in (0,1)$ be given
Then there exists $\alpha > 0$ such that for $\phi(\alpha) := f(x^{(k)} + \alpha . p^{(k)})$ then:

$$\phi(\alpha) \leq \phi(0) + \alpha . c_1 . \phi'(0)$$

+) c_1 too large $\Rightarrow \alpha \ll \Rightarrow x^{(k+1)} \approx x^{(k)}$

c_1 too small $\Rightarrow$ small reduction of f

+) Algorithm: - choose $\alpha_1 > 0$; $\rho, c_1 \in (0,1)$

Simple line-search algorithm

set $\alpha = \alpha_1$

repeat $\alpha \leftarrow \rho \alpha_1$ until $\phi(\alpha) \leq \phi(0) + \alpha . c_1 . \phi'(0)$

+) Improved Line-Search Algorithm:

- choose $\alpha_1 > 0$, $c_1 \in (0,1)$
- if $\phi(\alpha_0) \leq \phi(0) + \alpha_0 c_1 \phi'(0)$ STOP

  ELSE, $\alpha \in (0, \alpha_0)$ through quadratic interpolation

$$\alpha_1 = - \frac{\phi'(0) . \alpha_0^2}{2[\phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0]}$$

- if $\phi(\alpha_1) \leq \phi(0) + \alpha_1 c_1 \phi'(0)$ STOP

  else, find better $\alpha \in (0, \alpha_0)$ through cubic interpolation
- repeat cubic interpolation until satisfied

+) Wolfe Conditions

f continuously differentiable, $p^{(k)}$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$

Then There exists $\alpha > 0$ such that

$$\begin{cases} \phi(\alpha) \leq \phi(0) + \alpha . c_1 \phi'(0) \\ \phi'(\alpha) \geq c_2 . \phi'(0) \quad \text{slope condition} \end{cases}$$

$\Rightarrow$ guarantee minimum step length

The purpose is that to not take too small of a step (compared with Armijo conditions)

+) Steepest Descent:

choose $x^{(0)}$
for $k = 0, 1, \ldots$
  if $\|\nabla f(x^{(k)})\| \leq \varepsilon$ stop, else
    set $p^{(k)} = -\nabla f(x^{(k)})$
    determine the step length α_k
  set $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$
end for

+) Newton-Raphson Method:

Quadratic approximation of f at $x^{(k+1)}$

$$p^{(k)} = -\left[\nabla^2 f(x^{(k)})\right]^{-1} \nabla f(x^{(k)})$$

$$x^{(k+1)} = x^{(k)} + p^{(k)}$$

not globally convergence

(+) locally quadratic convergence, if $x^{(k)}$ is close to x^*

(-) need 2^{nd} derivative & inversion of matrix

if f is quadratic, algorithm converges in 1 step

- Problem: maybe
  - expensive
  - not invertable
  - quadratic is not the smart way

→ $S_k = 1$ cause we assume quadratic model & quadratic model has exact solution
we can still look for step-length solution, but not necessary

Complexity Analysis

- Zeroth order Oracle : Given x return value $f(x)$

  1st ______ : ______ $f(x)$, $\nabla f(x)$

  2nd ______ : ______ $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

- Analytical Complexity: smallest no. queries to solve problem P to accuracy ε

- Arithmetical ______ : ______ arithmatic operations ______

  (both work of oracle & work of method)

- Steepest Descent : worst case analytical complexity $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$

  Newton's method : $\mathcal{O}\left(\frac{1}{\varepsilon^{2-\tau}}\right)$

  $1 > \tau > 0$

(q, +) Inexact Newton Method:

Newton method: $H^{(k)} p^{(k)} = -g^{(k)}$

Now, we approximately solve $H^{(k)} p^{(k)} = -g^{(k)}$ with iterative method

$\Rightarrow$ Idea $H^{(k)} \approx B^{(k)}$

Modified Newton method
when $H^{(k)}$ ~~is~~ { singular or almost
not positive definite

$B^{(k)} = H^{(k)} + E^{(k)}$

with $E^{(k)} = \tau_k I$, $\tau_k \geq 0$ smartly chosen

converges to steepest descent for $\tau_k \to \infty$

- Davidon:

Inexact Newton $\neq$ Modified Newton $\neq$ Quasi Newton

+) Gauss - Newton Method.

$$\min_{x \in \mathbb{R}^2} f(x) = \min_{x \in \mathbb{R}^2} \frac{1}{2} \|\varepsilon(x)\|_2^2 = \min_{x \in \mathbb{R}^2} \frac{1}{2} \varepsilon(x)^T \varepsilon(x)$$

$$J(x) = \nabla \varepsilon(x) \quad \in \mathbb{R}^{m \times 2}$$

$$\Rightarrow \quad \nabla f(x) = J(x)^T \varepsilon(x)$$

$$\Rightarrow \quad \Delta f(x) = \nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} \underbrace{\varepsilon_j(x) \nabla^2 \varepsilon_j(x)}_{(\simeq 0)}$$

$$H(x) \simeq J^{(k)T} J^{(k)}$$

$$\Rightarrow \quad J^{(k)T} J^{(k)} p^{(k)} = - J^{(k)T} \varepsilon^{(k)}$$

Extra notes

- If $J^{(k)}$ has full rank, $p^{(k)}$ is always descent direction
- J is constant matrix $\Rightarrow$ linear model
- If $J^{(k)}$ is singular or almost singular $\Rightarrow$ $p^{(k)}$ is not reliable
  - $\Rightarrow$ converge very poorly
  - $\Rightarrow$ Quasi-Newtons are therefore more efficient
- Is a local method (to find local minimum not guarantee to find global minimum)

depends on initial value

+) Truste Region:

- Approximate objective function with Taylor expansion.

$$\cancel{f(x^{(k)})} \approx m^{(k)}(p) = f^{(k)} + g^{(k)T} \cdot p + \frac{1}{2} p^T B^{(k)} p$$

$$f^{(k)} = f(x^{(k)}), \quad g^{(k)} = \nabla f(x^{(k)}), \quad B^{(k)} \text{ is symmetric}$$

$$x^{(k+1)} = x^{(k)} + p^{(k)}$$

$$\Rightarrow p^{(k)} = \underset{p \in \Delta}{\operatorname{argmin}}\ m^{(k)}(p)$$

Δ ↳ trust region

from unconstrained opt problem
⇓ to
a constrained optimization problem

when Δ is closed $\mathcal{S}$.
$\Leftrightarrow$ always $\exists\ p^{(k)}$ optimal $\in \Delta$

Can be solved with
- Lagrange
- SQP (Sequential Quadratic Programming

$$\Delta = \{ p \in \mathbb{R}^n \mid \|p\|_2^2 \leq \delta \} \quad \text{with } \delta \geq 0$$

$$\rho_k = \frac{f(x^{(k)}) - f(x^{(k)} + p^{(k)})}{m^{(k)}(0) - m^{(k)}(p^{(k)})} = \frac{\text{actual reduction}}{\text{predict preduction}}$$

If
- $\rho < \frac{1}{4} \Rightarrow$ contract Δ: $\delta := \frac{1}{2}\delta$
- $\begin{cases} \rho > \frac{3}{4} \\ |p_k| = \delta \end{cases} \Rightarrow$ extend Δ: $\delta := 2\delta$

- Strategies for efficient solution to find $p^{(k)}$ in Δ
  - The Cauchy point: minimum along steepest descent $-g^{(k)}$ SLOW
  - Dogleg method: applicable when $B^{(k)}$ is positive definite
    converge very fast (super linear)
  - Steihaug's approach for large sparse matrixes

- Dogleg method
  - large δ
  - small δ
  - intermediate δ

Extra Notes

- Gauss-Newton

Also try to deal with H_k^{-1}, Gauss limit proplem $Q(\underline{p})$ to only have squares of m functions

$$Q(\underline{p}) = \sum_{i=1}^{m} q_i(p)^2 = |\underline{q}(\underline{p})|^2$$

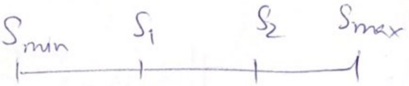
$$\Rightarrow \underline{p}_{k+1} = \underline{p}_k - (J_k^T J_k)^{-1} J_k^T \underline{q}(\underline{p}_k)$$

+) Choose s_k

- Pick $\varepsilon > 0$ small enough

$s_{min} = 0$, $s_{max} > s_{min}$ big enough

- Let $s_{min} < s_1 < s_2 < s_{max}$



$\hat{Q}(s_1) < \hat{Q}(s_2) \Rightarrow s_{max} = s_2$

else $s_{min} = s_1$

If $|s_{max} - s_{min}| < \varepsilon$ pick s_k between $[s_{min} \; s_{max}]$

else repeat

- Golden - section search:

$$\frac{s_1 - s_{min}}{s_{max} - s_1} = \frac{s_{max} - s_1}{s_{max} - s_{min}} \quad \& \quad \frac{s_{max} - s_2}{s_2 - s_{min}} = \frac{s_2 - s_{min}}{s_{max} - s_{min}}$$

Armijo is more commen (professor's thought)

But some people would go for Golden-section search

$Q(\underline{p})$ with $\underline{p} = (p_1 \ldots p_n)^T$

$\mathbb{P} \subseteq \mathbb{R}^n$

$\underline{p}^* \in \mathbb{P}$ such $Q(\underline{p}^*) = \min\limits_{\underline{p} \in \mathbb{P}} Q(\underline{p})$ $\qquad x^* = \min\limits_{x \in \mathbb{D}} f(x)$

$\mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) \leq d_i$ with $d_i \in \mathbb{R}$, $i = 1, 2, \ldots n \}$

Unconstrained problem

+) Line search $\quad \underline{p}^{(k+1)} = \underline{p}^{(k)} + s \cdot \underline{h}^{(k)}$

- Steepest descent (gradient descent)

$$\underline{h}_k = -\text{grad } Q(\underline{p})\Big|_{\underline{p}=\underline{p}_k} = -\left(\frac{\partial Q}{\partial \underline{p}}\right)^T\Bigg|_{\underline{p}=\underline{p}_k}$$

$$\| Q(\underline{p}_{k+1}) - Q(\underline{p}_k) \| < \varepsilon \Rightarrow \underline{p}^* \approx \underline{p}_{k+1}$$

we want

~~$f(p_k) = \text{grad } Q(p_k) = 0$~~

$f(p^*) = \text{grad } Q(p^*) = 0$

$f(p^*) \approx f(p_k) + f'(p_k)(p^* - p_k)$

only "=" when

$f(p_k) + f'(p_k)(p_{k+1} - p_k)$

$\Rightarrow p_{k+1} = p_k - f'(p_k)^{-1} \cdot f(p_k)$

- Newton-Raphson

$$\underline{p}_{k+1} = \underline{p}_k - H_k^{-1} f(\underline{p}_k) = \underline{p}_k - H_k^{-1} \cdot \text{grad } Q(\underline{p}_k)$$

$$H_k = \frac{\partial^2 Q}{\partial \underline{p}^2}\bigg|_{\underline{p}=\underline{p}_k} = \left.\begin{bmatrix} \frac{\partial^2 Q}{\partial p_1 \partial p_1} & \cdots & \frac{\partial^2 Q}{\partial p_1 \partial p_n} \\ \\ \frac{\partial^2 Q}{\partial p_n \partial p_1} & & \frac{\partial^2 Q}{\partial p_n \partial p_n} \end{bmatrix}\right|_{\underline{p} = \underline{p}_k}$$

- quasi Newton

Newton-Raphson problem: has to inverse H_k at every steps

Now, we estimate $H_k^{-1} \approx B_k =$

some formula:
- DFP (Davidon-Fletcher-Powell)
- BFGS (Broyden-Fletcher-Goldfarb-Shanno)
- SR1 (Symmetric rank one)
- Broyden

$$\underline{p}_{k+1} = \underline{p}_k - s_k \cdot B_k \text{ grad } Q(\underline{p}_k)$$

- Newton:

Try to set grad $Q(p^*) = 0 = f(p^*)$

Approximate $f(p^*)$ around p_k:

$$0 = f(p^*) \simeq f(p_k) + f'(p_k)(p^* - p_k)$$
$$= f(p_k) + H_k (p^* - p_k) \quad (\text{call } f'(p_k) \text{ as } H_k)$$

If ~~we~~ it's just $\simeq$, then must exist p_{k+1} so ~~we~~ will have equality

$$0 = f(p_k) + H_k (p_{k+1} - p_k)$$

$\Rightarrow$ Sequence $p_{k+1} = p_k - H_k^{-1} \cdot f(p_k)$

- quasi Newton:

$$f(p_{k+1}) \simeq f(p_k) + H_k d_k = f(p_k) + f'(p_k)(p_{k+1} - p_k)$$

$$f(p_k) = \text{grad } Q(p_k)$$

From approximation of $Q(p_k)$:

$$Q(p_{k+1}) = Q(p_k + d_k) \simeq Q(p_k) + \text{grad } Q(p_k)^T d_k + \frac{1}{2} d_k^T H_k d_k$$

$$\Rightarrow \text{grad } Q(p_{k+1}) \simeq \text{grad } Q(p_k) + H_k d_k$$

$$\Rightarrow d_k \simeq B_k y_k \left(= H_k^{-1} \cdot (\text{grad } Q(p_{k+1}) - \text{grad } Q(p_k))\right)$$

+) Trust region

$$Q(\underline{p}) \approx Q(\underline{p_k}) + \underline{b_k}^T \underline{d} + \frac{1}{2} \underline{d}^T H_k \underline{d} = Q(\underline{p_k}) + \phi(\underline{d})$$

in which:

$\phi(\underline{d}) = b_k^T d + \frac{1}{2} d^T H_k \underline{d}$, $\underline{d} = \underline{p} - \underline{p_k}$, $b_k = \text{grad } Q(p_k)$

$$H_k = \left. \frac{\partial^2 Q}{\partial p} \right|_{\underline{p = p_k}}$$

$$\underline{d_k} = \underset{\underline{d} \in \Delta}{\arg\min} \ \phi(\underline{d})$$

$\underline{d} \in \Delta \rightarrow$ trust region

a constrained optimization problem

Δ is closed & giới nội?

$\Leftrightarrow$ always $\exists$ $\underline{d_k}$ optimal $\in \Delta$

can be solve with
- Lagrange
- SQP Sequential quadratic Programming

$$\Delta = \{ \underline{d} \in \mathbb{R}^n \mid |\underline{d}|^2 \leq \delta \} \quad \text{with } \delta \geq 0$$

$$\rho = \frac{Q(p_k) - Q(p_{k+1})}{\phi(0) - \phi(d_k)}$$

If $\rho < \frac{1}{4} \Rightarrow \delta := \frac{1}{2} \delta$

$\rho > \frac{3}{4}$ and $|\underline{d_k}| = \delta \Rightarrow \delta := 2\delta$

} To extend / contract trust region Δ

ANO

Constrained Problem

L5)

From Constrained Problem to Unconstrained Problem

Barrier function method:

- Given: $\underline{p}^* = \arg\min_{\underline{p} \in \mathbb{P}} Q(\underline{p})$ with $\mathbb{P} = \{\underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) \leq 0, i=1,..m\}$

- Barrier func $B(\underline{p})$: as $\underline{p}$ close to edge of $\mathbb{P}$, the $B(\underline{p})$ is greater

  Example: $B(\underline{p}) = \max_i b(g_i(\underline{p}))$ or $B(\underline{p}) = \sum_{i=1}^{m} b(g_i(\underline{p}))$

  in which $b(z) = |z|^{-k}$, $k>0$ or $b(z) = -\ln|z|$

- New objective func: $\widehat{Q}(\underline{p}, \lambda) = Q(\underline{p}) + \lambda . B(\underline{p})$

$\Rightarrow$ Find $\underline{\hat{p}}^*(\lambda) = \arg\min \widehat{Q}(\underline{p}, \lambda)$

$\lim_{\lambda \to 0} \underline{\hat{p}}^*(\lambda) = \underline{p}^*$

Penalty function method:

$P(\underline{p}) = \begin{cases} 0 & \text{if } \underline{p} \in \mathbb{P} \\ >0 & \underline{p} \notin \mathbb{P} \end{cases}$

Ex: $P(\underline{p}) = \frac{1}{2} \sum_{i=1}^{m} \left(\max\{0, g_i(\underline{p})\}\right)^2$

- $\widehat{Q}(\underline{p}, \lambda) = Q(\underline{p}) + \lambda P(\underline{p})$

- $\underline{\hat{p}}^*(\lambda) = \arg\min \widehat{Q}(\underline{p}, \lambda)$

- If $\underline{\hat{p}}^*(\lambda) \in \mathbb{P}$ then $\underline{\hat{p}}^*(\lambda) = \underline{p}^*$

  and always that $\lim_{\lambda \to \infty} \underline{\hat{p}}^*(\lambda) = \underline{p}^*$

Lagrange's method

- $\underline{p}^* = \arg\min_{\underline{p} \in \mathbb{P}} Q(\underline{p})$ with $\mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) = 0,\ i = 1, 2, \dots m \}$

only with equality constraints

- $\hat{\underline{p}} = col(\underline{p}, \underline{\lambda}) \in \mathbb{R}^{n+m}$; $\underline{\lambda} = [\lambda_1, \lambda_2 \dots \lambda_m]^T$

- $L(\hat{\underline{p}}) = L(\underline{p}, \underline{\lambda}) = Q(\underline{p}) + \sum_{i=1}^{m} \lambda_i g_i(\underline{p}) = Q(\underline{p}) + \underline{\lambda}^T \underline{g}(\underline{p})$

$$\Rightarrow \underline{0}^T = \left.\frac{\partial L}{\partial \hat{\underline{p}}}\right|_{\hat{\underline{p}}^*} \Leftrightarrow \begin{cases} \left.\left(\frac{\partial Q}{\partial \underline{p}} + \underline{\lambda}^T \frac{\partial \underline{g}}{\partial \underline{p}}\right)\right|_{\underline{p}^*, \underline{\lambda}^*} = \underline{0}^T \\ \underline{g}(\underline{p}^*) = \underline{0} \end{cases}$$

Solve nonlinear system of equations

Kuhn-Tucker Condition:

$L(\underline{p}^*, \underline{\lambda}) \leq L(\underline{p}^*, \underline{\lambda}^*) \leq L(\underline{p}, \underline{\lambda}^*) \ \forall \underline{p}, \underline{\lambda}$

Problem: $\underline{p}^* = \arg\min_{\underline{p} \in \mathbb{P}} Q(\underline{p})$ with $\mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) \leq 0,\ i = 1, 2, \dots m \}$

in equality constraints

$L(\underline{p}, \underline{\lambda}) = Q(\underline{p}) + \underline{\lambda}^T \underline{g}(\underline{p})$ with $\lambda_i \geq 0 \ \forall i$

$\Rightarrow$ – Neccessary condition: If $(\underline{p}^*, \underline{\lambda}^*)$ is saddle point of $L(\underline{p}, \underline{\lambda})$ then $\underline{p}^*$ is optimal solution of $Q(\underline{p})$

– Sufficient condition: If $Q(\underline{p})$, $g_i(\underline{p})$ are convex

$\mathbb{P}$ has at least 1 point

Then $(\underline{p}^*, \underline{\lambda}^*)$ is saddle point iff $\underline{p}^*$ is optimal solution of $Q(\underline{p})$

$\Rightarrow$ Kuhn-Tucker's Method:

+) Nece. conds for $(\underline{p}^*, \underline{\lambda}^*)$ to be saddle point of $L(\underline{p}, \underline{\lambda})$ are

$$\begin{cases} grad_{\underline{p}} L(\underline{p}^*, \underline{\lambda}^*) = \underline{0} \\ grad_{\underline{\lambda}} L(\underline{p}^*, \underline{\lambda}^*) \leq \underline{0} \quad \text{(all neg-element vector)} \\ (\underline{\lambda}^*)^T grad_{\underline{\lambda}} L(\underline{p}^*, \underline{\lambda}^*) = 0 \end{cases}$$

+) Suff. cond.

If $Q(\underline{p})$, $\underline{g}(\underline{p})$ convex

$\mathbb{P}$ has at least 1 point

$\Rightarrow$ also sufficient condition

+) <u>Equality constraints</u>

- In order to not exist p such $\begin{cases} \nabla f(x)^T p < 0 \\ \nabla c(x)^T p = 0 \end{cases}$ $\Rightarrow$ Set $\nabla f(x) \mathbin{//} \nabla c(x)$ $(\Rightarrow \nabla f(x)^T p = \nabla c(x)^T p = 0)$

Thus we want $\nabla f(x) + \lambda \nabla c(x) = 0$

$\Rightarrow$ 1st-order necessary optimality conditions $\begin{cases} \nabla_x L(f, \lambda) = \nabla f(x) + \lambda \nabla c(x) = 0 & \text{stationarity} \\ \nabla_\lambda L(f, \lambda) = \quad c(x) = 0 & \text{primal feasibility} \end{cases}$

- Sign of $c, \nabla c, \lambda$ is arbitrary

+) <u>In-equality constraints</u>

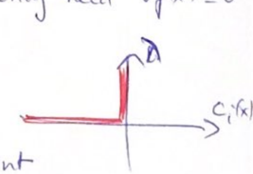
- 2 case: $\left[\begin{array}{l} \text{point lies inside feasible set:} \\ \quad \text{need } \nabla f(x) = 0 \text{ so that } \nexists\ \nabla f(x)^T p < 0 \\ \text{point lies on edge feasible set: } c(x) = 0 \end{array}\right.$

$\Leftrightarrow$ no p ~~such~~ $\begin{cases} \nabla f(x)^T p < 0 \\ \nabla c(x)^T p \leq 0 \end{cases}$ $\Leftrightarrow$ no p such $\begin{cases} \nabla f(x) + \lambda \nabla c(x) = 0 \\ \lambda \geq 0 \end{cases}$

- $\begin{cases} \nabla f(x) + \lambda \nabla c(x) = 0 & \text{stationarity condition} \\ c(x) \leq 0 & \text{primal feasibility} \\ \lambda . c(x) = 0 & \text{complementarity condition} \\ \lambda \geq 0 & \text{dual feasibility} \end{cases}$

- Together with primal & dual feasibility condition, complementary slackness is equipvalent to if-else statements:

$\left[\begin{array}{lll} \text{constraints inactive} & c_i(x) < 0 \Rightarrow \lambda_i = 0 & \Leftrightarrow \text{only need } \nabla f(x) = 0 \\ \text{constraints active} & c_i(x) = 0 \Rightarrow \lambda_i \geq 0 & \end{array}\right.$



- Complementarity Slackness (CS) is non-smooth constraint

+) <u>Active set</u>: the set of the active constraints

$$A(x) = E \cup \{ i \in I \mid c_i(x) = 0 \}$$

1st-order necessary

+) KKT Conditions (Karush - Kuhn - Tucker)

Let $x^* \in \mathbb{R}^n$ be local minimizer

Assume $\nabla c_i(x^*)$, $i \in A(x^*)$ are linearly independent

Then $\exists\ \lambda_i^*$, $i \in E \cup I$:

$$\begin{cases} \nabla f(x^*) + \sum_{i \in E \cup I} \lambda_i^* \nabla c_i(x^*) = 0 & \text{stationarity} \\ c_i(x^*) = 0, \ \forall i \in E & \text{primal feasibility} \\ c_i(x^*) \leq 0, \ \forall i \in I & \\ \lambda_i^* \geq 0, \ \forall i \in I & \text{dual feasibility} \\ \lambda_i^* c_i(x^*) = 0, \ \forall i \in I & \text{complementary slackness} \end{cases}$$

+) Sufficient Optimality Conditions:

(2nd order)

Let $x^* \in \mathbb{R}^n$ and x^*, λ^* satisfy KKT conditions

If $p^T \nabla_{xx} L(x^*, \lambda^*) p > 0 \quad \forall p \neq 0$

$\nabla c_i(x)^T p = 0, \ \forall i \in E$

$\nabla c_i(x)^T p = 0, \ \forall i \in I \cap A(x^*)$ with $\lambda_i^* > 0$

$c_i(x)^T p \geq 0, \ \forall i \in I \cap A(x^*)$ with $\lambda_i^* = 0$

Then $x^* \in \mathbb{R}^n$ is local minimizer of f

convex

+) Sensitivity analysis:

how important is each constraints

whether perturbation of (in)active constraints has impact on optimal value

ANO

Linear Programming

L6)

Problem: $\underline{p}^* = \arg\min_{p \in \mathbb{P}} G(\underline{p}) = \underline{a}^T \underline{p}$ $\quad a = [a_1, .., a_n]^T$ (2.102)

Linear Programming

obj func $G(\underline{p})$ is linear

constraints $g_i(\underline{p})$ are linear

$$\mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid \underline{c}_i^T \underline{p} \le d_i, \; i = 1, 2, .. m$$
$$p_j \ge 0, \; j = 1, 2, .. n \}$$

$$\underline{c}_i^T = (c_{i1}, c_{i2}, .. c_{in}) \quad \text{and } d_i \in \mathbb{R}, \; i = 1, 2, .. m$$

⇒ **Standard form:**

$$\hat{\underline{p}}^* = \arg\min_{\underline{p} \in \mathbb{P}} (\hat{\underline{a}}^T \hat{\underline{p}}) \quad \text{with } \hat{\mathbb{P}} = \{ \hat{\underline{p}} \in \mathbb{R}^{n+m} \mid \hat{\underline{c}}_i^T \hat{\underline{p}} = d_i, \; i = 1, 2, .., m$$
$$p_j \ge 0, \; j = 1, 2, .. n+m \}$$
$$d_i \ge 0, \; i = 1, 2, .., m$$

If $d<0$ ⇒ multiply with (-1)

$$\hat{\underline{p}} = [p_1, \dots, p_n, p_{n+1}, .. p_{n+m}]^T$$

$$\hat{\underline{a}}^T = [a^T, \underbrace{0, .., 0}_{m \text{ element}}]$$

$$\hat{c}_i^T = [\underline{c}_i^T, 0, 0, \underset{\substack{\uparrow \\ n+i \text{ element}}}{1}, .., 0] \Leftrightarrow \underline{c}_i^T \underline{p} + p_{n+i} = d_i, \; p_{n+i} \ge 0$$

⇔ For convenience: $\hat{\mathbb{P}} = \{ \hat{\underline{p}} \in \mathbb{R}^{n+m} \mid C \hat{\underline{p}} = \underline{d}, \; \underline{d} \ge \underline{0}, \hat{\underline{p}} \ge \underline{0} \}$

$$C \in \mathbb{R}^{m \times (n+m)}, \quad C = \begin{bmatrix} \hat{c}_1^T \\ \hat{c}_2^T \\ \vdots \\ \hat{c}_m^T \end{bmatrix}$$

$\Rightarrow$ Simplex method:

the optimal point must be corner (or edge)

so simplex method search the corners

$$\underline{d} = C\hat{\underline{f}} = S_k f_{SP} + E_k f_{KSP}$$

$$\Rightarrow f_{SP} = S_k^{-1}(\underline{d} - E_k f_{KSP})$$

Replace into $Q(\hat{\underline{f}}) = \hat{\underline{a}}^T \hat{\underline{f}}$

$$Q(\hat{\underline{f}}) = \hat{\underline{a}}^T \hat{\underline{f}} = \hat{\underline{a}}^T (S_k^{-1}(\underline{d} - E_k f'_{KSP}), f_{KSP})$$
$$= Q'(f_{KSP})$$

- Tabular form

Initial basis, because just 1 in 1 row, every other row is 0

$Q(f)$	x_1	..	x_n	x'_{n+1}	..	x'_{n+m}	RHS
Objective func	?	? ..	?	0	..	0	
x'_{n+1}				1		0	
$\vdots$				0	$\ddots$	0	
x'_{n+m}				0		1	

constraints, also basis variable

- Find the most negative, try to add to basic variable by turning all other row coefficients to 0 (choose row with min rate $\frac{RHS}{\text{cur. coeff.}}$ $\rightarrow$ that's the constraint limit the most the new variables

- Untill all coefficients of first row (the obj. func row) is positive

all $\lambda_{I,N} \geq 0$

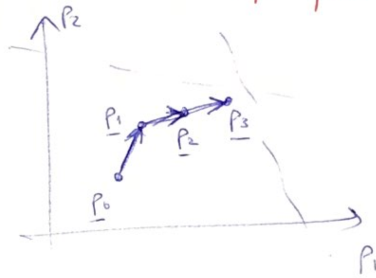
but only need to check for variables not in the basic variable because for basic variables, the coefficients in obj func row are 0

- Also check for Big M's Method

7) Interior Point method for convex opt problem

: for both linear convex opt. prob. or nonlinear convex opt. prob.

not just linear convex opt prob



→ Using block function / barrier function:

- Consider nonlinear convex opt problem

$$\underline{p}^* = \underset{\underline{p} \in \mathbb{P}}{\operatorname{argmin}}\ Q(\underline{p})$$

with $\mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) \leq 0,\ i = 1, 2, \dots, m\}$

$Q(\underline{p})$, $g_i(\underline{p})$ are convex

- $B(\underline{p})$ as convex barrier function

⇒ Set $\hat{Q}(\underline{p}, \mu) = Q(\underline{p}) + \mu \, B(\underline{p})$ with given small $\mu > 0$

$\hat{\underline{p}}^*(\mu) = \operatorname{argmin} \hat{Q}(\underline{p}, \mu)$ and $\lim_{\mu \to 0} \hat{\underline{p}}^*(\mu) = \underline{p}^*$ (Final result $\hat{\underline{p}}^*$ as approximation of $\underline{p}^*$

- Find $\hat{\underline{p}}^*(\mu)$ from $\dfrac{\partial \hat{Q}(\underline{p}, \mu)}{\partial \underline{p}} = \underline{0}^T \Leftrightarrow \underline{f}(\underline{p}, \mu) = \underline{0}$

$= \operatorname{grad} \hat{Q}(\underline{p}, \mu)$

⇒ use Newton - Raphson for unconstrained prob

⇒ Algorithm:

- Choose barrier func $B(\underline{p})$ convex and $\mu > 0$ small enough
  → $\hat{Q}(\underline{p}, \mu)$ for unconstrained problem
- Calculate $\underline{f}(\underline{p}, \mu)$
- Choose $\underline{p}_0 \in \mathbb{P}$, $k = 0$
- Calculate $H_k = \left. \dfrac{\partial \underline{f}(\underline{p}, \mu)}{\partial \underline{p}} \right|_{\underline{p}_k}$ (Hessian matrix of $\hat{Q}(\underline{p}, \mu)$ at $\underline{p}_k$)
- Calculate $\underline{p}_{k+1} = \underline{p}_k - H_k^{-1} \underline{f}(\underline{p}_k)$
- Stop when $| \hat{Q}(\underline{p}_{k+1}, \mu) - \hat{Q}(\underline{p}_k, \mu) | < \varepsilon$ with chosen $\varepsilon > 0$ very small or when $k > K$ with great K

+) Using Lagrange function

$$p^* = \underset{p \in \mathbb{P}}{\arg\min}\ Q(p) \quad \text{with } \mathbb{P} = \{ p \in \mathbb{R}^n \mid g_i(p) \le 0,\ i = 1, 2, \dots m \}$$

add m new variables: $p_{n+1}, \dots, p_{n+m} \ge 0$

so that $g_i(p) \le 0 \Leftrightarrow g_i(p) + p_{n+i} = 0$, $i = 1, 2, \dots m$

$\Rightarrow$ new problem: $\hat{p}^* = \underset{\hat{p} \in \hat{\mathbb{P}}}{\arg\min}\ Q(\hat{p})$ with $\hat{\mathbb{P}} = \{ \hat{p} \in \mathbb{R}^{n+m} \mid \hat{g}(\hat{p}) = 0 \text{ and } p' \ge 0 \}$

$$\hat{p} = col(p^T, p') \ ; \ p' = [p_{n+1} \dots p_{n+m}]^T$$

- To remove inequality constraints: use barrier function:

$$\hat{Q}(\hat{p}, \mu) = Q(\hat{p}) + \mu B(p') \qquad \mu \text{ small } > 0$$

$$\Rightarrow \hat{p}^* = \underset{\hat{p} \in \hat{\mathbb{P}}}{\arg\min}\ \hat{Q}(\hat{p}, \mu) \quad \text{with } \hat{\mathbb{P}} = \{ \hat{p} \in \mathbb{R}^{n+m} \mid \hat{g}(\hat{p}) = 0 \}$$

. Use Newton method for nonlinear, equality constraint problem

Used as basis for other methods
Cause we can approximate with Taylor expansion, get $\Rightarrow$ a sub quadratic opt. prob.

Quadratic Programming: constrained problem

quadratic objective func

linear $\begin{cases} \text{equality} \\ \text{inequality} \end{cases}$ constraints

$$\min_{\underline{p} \in \mathbb{P}} Q(\underline{p}) = \frac{1}{2} \underline{p}^T A \underline{p} + \underline{b}^T \underline{p} \quad \text{with } \mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid \underline{c}_i^T \underline{p} = d_i, \ i = 1, 2, \dots m' ; \ \underline{c}_j^T \underline{p} \leq d_j, \ j = m'+1, \dots m \}$$

$$A = A^T \geqslant 0 \qquad m \leq n$$

+) Only linear equality constraints: $m' = m$

$$\Leftrightarrow \mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid \underline{c}_i^T \underline{p} = d_i \quad i = 1, 2, \dots m \}$$
$$= \{ \underline{p} \in \mathbb{R}^n \mid g(\underline{p}) = C\underline{p} - \underline{d} = \underline{0} \} \longrightarrow C = \begin{bmatrix} c_1^T \\ c_2^T \\ \vdots \\ c_m^T \end{bmatrix}$$

- Use Lagrange func and KKT conditions:

$$\begin{cases} \dfrac{\partial Q(\underline{p})}{\partial \underline{p}} \underline{p}^* + \dfrac{\partial g(\underline{p})}{\partial \underline{p}} \underline{\lambda}^* = \underline{0} \\ C\underline{p}^* - \underline{d} = \underline{0} \end{cases} \Leftrightarrow \underbrace{\begin{bmatrix} A & C^T \\ C & 0 \end{bmatrix}}_{D} \cdot \begin{bmatrix} \underline{p}^* \\ \underline{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\underline{b} \\ \underline{d} \end{bmatrix}$$

very cool because we can go directly to solution

$$\Rightarrow \begin{bmatrix} \underline{p}^* \\ \underline{\lambda}^* \end{bmatrix} = D^{-1} \begin{bmatrix} -\underline{b} \\ +\underline{d} \end{bmatrix} = \begin{bmatrix} A^{-1}C^T(CA^{-1}C^T)^{-1}(CA^{-1}\underline{b} + \underline{d}) - A^{-1}\underline{b} \\ -(CA^{-1}C^T)^{-1}(CA^{-1}\underline{b} + \underline{d}) \end{bmatrix}$$

+) Both linear equality and inequality constraints:

$$\mathcal{A}_E = \{1, 2, \dots m'\} \text{ and } \mathcal{A}_I = \{m'+1, \dots, m\}$$
$$\mathcal{A}_k = \{1, 2, \dots m', \dots\}$$

some inequality from $\mathcal{A}_I$

$$\Rightarrow \mathcal{A}_E \subset \mathcal{A}_k \subseteq \mathcal{A}_E \cup \mathcal{A}_I \quad \forall k$$

$$\Rightarrow \mathbb{P}_k = \{ \underline{p} \in \mathbb{R}^n \mid \underline{c}_l^T \underline{p} = d_l, \ l \in \mathcal{A}_k \} \text{ a border part of } \mathbb{P}$$

- At step k: $\underline{p}_{k+1} = \underline{p}_k + s_k \underline{h}_k$, assume $s_k = 1$

$$Q(\underline{p}_{k+1}) = \frac{1}{2} (\underline{p}_k + \underline{h}_k)^T A (\underline{p}_k + \underline{h}_k) + \underline{b}^T (\underline{p}_k + s_k \underline{h}_k) = \frac{1}{2} \underline{h}_k^T A \underline{h}_k + \hat{\underline{b}}^T \underline{h}_k + \underline{e}_k$$

with $\underline{e}_k = \frac{1}{2} \underline{p}_k^T A \underline{p}_k + \underline{b}^T \underline{p}_k$ and $\hat{\underline{b}} = A\underline{p}_k + \underline{b}$

At each step, solve a QP with only linear equality constraints

$$\Rightarrow \underline{h}_k = \underset{\underline{h}_k \in \mathbb{P}'_k}{\operatorname{argmin}} \ \hat{Q}(\underline{h}) = \frac{1}{2} \underline{h}_k^T A \underline{h}_k + \hat{\underline{b}}^T \underline{h}_k \quad \text{with } \mathbb{P}'_k = \{ \underline{h}_k \in \mathbb{R}^n \mid \underline{c}_l^T \underline{h}_k = 0, \ l \in \mathcal{A}_k \}$$

$$\Rightarrow \begin{bmatrix} \underline{h}_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} A & C_k^T \\ C_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\underline{b}} \\ \underline{0} \end{bmatrix}$$

- Need to check if $p_{k+1} \in \mathbb{P}$
- If we have $p_k \in \mathbb{P}_k$ thus $p_{k+1} \in \mathbb{P}_k$
- If $\underline{h}_k \neq 0$
  $\exists\ l \in \mathcal{A}_I \setminus \mathcal{A}_k$ that $\underline{c}_l^T \underline{h}_k \leq 0$ & $\underline{c}_l^T p_k \leq \underline{d}_l$ $\Rightarrow \underline{c}_l^T p_{k+1} \leq \underline{d}_l \quad \forall s_k$ ($\Rightarrow$ then pick $s_k = 1$)
- If $\underline{h}_k \neq 0$
  $\exists\ l \in \mathcal{A}_I \setminus \mathcal{A}_k$ that $\underline{c}_l^T \underline{h}_k \geq 0$
  in order for $\underline{c}_l^T p_{k+1} \leq d_l \Leftrightarrow s_k < \dfrac{d_l - \underline{c}_l^T p_k}{\underline{c}_l^T \underline{h}_k}$

⊗ Algorithm QP for both linear equality & inequality constraints

+) $\mathcal{A}_0 = \mathcal{A}_E = \{1, 2, \dots m'\}$, $k = 0$
   $p_0 = \arg\min Q(p) = \frac{1}{2} p^T A p + \underline{b}^T p \qquad \mathbb{P}_E = \{ p \in \mathbb{R}^n \mid \underline{c}_i^T p = d_i,\ i \in \mathcal{A}_E \}$
   $\Rightarrow p_0 = A^{-1} C_E^T (C_E A^{-1} C_E^T)^{-1} C_E A^{-1} \underline{b} - A^{-1} \underline{b}$

+) Find $\underline{h}_k, \lambda_k$ (from p_k to $\hat{\underline{b}}$)

+) If $\underline{h}_k \neq 0$
   - $s_k = \min\{1, s^*\}$; $s^* = \min\limits_{i \in I} \dfrac{d_i - \underline{c}_i^T \underline{h}_k}{\underline{c}_i^T \underline{h}_k}$; $I = \{ i \in \mathcal{A}_E \cup \mathcal{A}_I \mid i \notin \mathcal{A}_k,\ c_i^T \underline{h}_k > 0 \}$
   - If $s_k = s^* \Rightarrow \mathcal{A}_{k+1} = \mathcal{A}_k \cup \{l\}$
     else $\mathcal{A}_{k+1} = \mathcal{A}_k$

+) If $\underline{h}_k = 0$
   - $\lambda_{k,i}^* = \arg\min\limits_{j \in \mathcal{A}_I \cap \mathcal{A}_k} \lambda_{k,j}$
   - If $\lambda_{k,i}^* \geq 0 \Rightarrow p^* = p_k \ (\forall i)$
     else $\mathcal{A}_{k+1} = \mathcal{A}_k \setminus \{i\}$

8) ⊛ Sequential Quadratic Programming: constrained problem.

$$\underline{p}^* = \underset{\underline{p} \in \mathbb{P}}{\arg\min}\ Q(\underline{p}) \quad \text{with } \mathbb{P} = \{ \underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) = 0,\ i = 1, 2, \dots m' ;\ g_j(\underline{p}) \le 0,\ j = m'+1 \dots m \}$$

nonlinear obj. func — nonlinear constraints

⊛ General idea: transform to Quadratic Programming (by approximation)

+) At step k: $\underline{p}_{k+1} = \underline{p}_k + \underline{h}_k \in \mathbb{P}$ that $Q(\underline{p}_{k+1}) < Q(\underline{p}_k)$

Use Taylor expansion

$$Q(\underline{p}_{k+1}) = Q(\underline{p}_k + \underline{h}_k) \approx Q(\underline{p}_k) + \underline{b}_k^T \underline{h} + \frac{1}{2} \underline{h}^T H \underline{h} = Q(\underline{p}) + \phi(\underline{h})$$

$$\phi(\underline{h}) \Rightarrow \min_{\underline{h}} \iff Q(\underline{p}_{k+1}) \to \min_{\underline{p}}$$

$$\iff \underline{h}_k = \underset{\underline{h} \in \mathbb{P}_k}{\arg\min}\ \phi(\underline{h}) \quad \text{with } \phi(\underline{h}) = \frac{1}{2} \underline{h}^T H_k \underline{h} + \underline{b}_k^T \underline{h};\ \underline{b}_k = \text{grad } Q(\underline{p}_k)$$

$$H_k = \left.\frac{\partial^2 Q}{\partial \underline{p}}\right|_{\underline{p} = \underline{p}_k}$$

Approximate { obj func; constraints

$$\mathbb{P}_k = \{ \underline{h} \in \mathbb{R}^n \mid \underline{c}_{ki}^T \underline{h} = d_i,\ i \in \mathcal{A}_E ;\ \underline{c}_{kj}^T \underline{h} \le d_j,\ j \in \mathcal{A}_I \}$$

$$\underline{c}_{kl}^T = \left.\frac{\partial g_l}{\partial \underline{p}}\right|_{\underline{p} = \underline{p}_k},\quad d_l = -g_l(\underline{p}_k)$$

$$\mathcal{A}_E = \{1, \dots m'\};\quad \mathcal{A}_I = \{m'+1, \dots m\}$$

→ SQP Algorithm

+) Set $\mathcal{A}_0 = \mathcal{A}_E = \{1, 2, \dots m'\}$, $k = 0$

Find $\underline{p}_0$ & $\underline{\lambda}_0$ from $\min_{\underline{p} \in \mathbb{P}_E} Q(\underline{p})$, $\mathbb{P}_E = \{\underline{p} \in \mathbb{R}^n \mid g_i(\underline{p}) = 0, i \in \mathcal{A}_E\}$

with Lagrange or Newton

+) Find $\phi(\underline{h})$, then find $(\underline{h}_k, \underline{\lambda}_k^h)$ with QP

+) Set $\underline{p}_{k+1} = \underline{p}_k + \underline{h}_k$ and $\underline{\lambda}_{k+1} = \underline{\lambda}_k^h$

+) Stop when $|Q(\underline{p}_{k+1}) - Q(\underline{p})| < \varepsilon$, $\varepsilon > 0$

or $k > K$

only

nonlinear obj func, equality constraints

Newton method

$$p^* = \underset{p \in \mathbb{P}}{\operatorname{argmin}}\ Q(p) \quad \text{with } \mathbb{P} = \{ p \in \mathbb{R}^n \mid g_i(p) = 0,\ i = 1, 2, \dots m' \}$$

Again, use approximation to transfer it to QP

$$- \quad \mathcal{L}(p, \underline{\lambda}) = Q(p) + \underline{\lambda}^T g(p) \qquad \text{with } g(p) = [g_1(p), \dots g_{m'}(p)]^T$$

$$\Rightarrow \quad \cancel{\mathcal{L}(p_{k+1}, \underline{\lambda}_{k+1}) \approx \mathcal{L}\, p_k}$$

approximate Lagrange func

$$\mathcal{L}(p, \underline{\lambda}) \approx \mathcal{L}(p_k, \underline{\lambda}_k) + \frac{1}{2} \hat{\underline{h}}^T \mathcal{H}_k \hat{h} + \underline{b}_k^T \hat{\underline{h}} \quad \text{and} \quad g(p) \approx -\underline{d}_k + C_k \hat{p}$$

$$= -\underline{d}_k + [C_k, \Theta]\, \hat{\underline{h}}$$

$$\hat{\underline{h}} = \begin{bmatrix} \hat{p} \\ \hat{\underline{\lambda}} \end{bmatrix} = \begin{bmatrix} p - p_k \\ \lambda - \lambda_k \end{bmatrix}; \quad \underline{b}_k^T = \left. \frac{\partial \mathcal{L}(p, \underline{\lambda})}{\partial(p, \underline{\lambda})} \right|_{p_k, \lambda_k};$$

$$\mathcal{H}_k = \left. \frac{\partial \mathcal{L}^2(p, \underline{\lambda})}{\partial(p, \underline{\lambda})} \right|_{p_k, \lambda_k}; \quad C_k = \left. \frac{\partial g}{\partial p} \right|_{p_k}, \quad \underline{d}_k = -g(p_k)$$

$$\Rightarrow \quad \hat{\underline{h}}_k = \underset{\underline{h} \in \mathbb{P}_k}{\operatorname{argmin}} \left(\frac{1}{2} \hat{\underline{h}}^T \mathcal{H}_k \hat{\underline{h}} + \underline{b}_k^T \hat{\underline{h}} \right) \quad \text{with } \mathbb{P}_k = \{ \hat{\underline{h}} \in \mathbb{R}^{n+m'} \mid \mathcal{C}_k \hat{\underline{h}} = \underline{d}_k \}$$

$$\mathcal{C}_k = [C_k, \Theta]$$

$\Rightarrow$ Algorithm: Newton method

+) Choose $(p_0, \underline{\lambda}_0)$ for $k = 0$

+) Calculate $\underline{b}_k^T, \underline{d}_k, \mathcal{H}_k, C_k \Rightarrow \mathcal{C}_k$

$$\Rightarrow \hat{\underline{h}}_k = \begin{bmatrix} \mathcal{H}_k^{-1} \mathcal{C}_k^T (\mathcal{C}_k \mathcal{H}_k^{-1} \mathcal{C}_k^T)^{-1} (\mathcal{C}_k \mathcal{H}_k^{-1} \underline{b}_k + \underline{d}_k) - \mathcal{H}_k^{-1} \underline{b}_k \\ -(\mathcal{C}_k \mathcal{H}_k^{-1} \mathcal{C}_k^T)^{-1} (\mathcal{C}_k \mathcal{H}_k^{-1} \underline{b}_k + \underline{d}_k) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} p_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} p_k \\ \lambda_k \end{bmatrix} + \hat{\underline{h}}_k$$

+) Stopping condition ...

also work with inequality constraint if we add variables s / barrier func / penalty

ANO

L6)

Linear Programming

- Problem: $\min_x c^T x$ $\quad x \in \mathbb{R}^n$

s.t. $a_i^T x - b_i = 0$, $i \in E$ $\iff$ $Ax = b$ $\quad A_{m \times n}$

$-x_i \leq 0$, $i \in \{1, \dots n\}$

- Assumptions: more variables than constraints $n > m$

rank $(A) = m$ $\quad$ $m \underbrace{\boxed{A}}^{n}$ $(m < n)$

- Feasible set: $\Omega = \{x \in \mathbb{R}^n \mid x \geq 0,\ Ax = b\}$ is a polytope

- $Ax = b \iff \begin{bmatrix} \underset{m \times m}{B} & \underset{m \times n}{N} \end{bmatrix} \begin{bmatrix} \overbrace{x_B}^{m \times 1} \\ \underbrace{x_n}_{m \times 1} \end{bmatrix} = b$

- Basic feasible point if we can chosen index set $|T(x)| = m$, $T(x) \in \{1, \dots m\}$

that $B := [a_i]_{i \in T(x)}$ is regular basis matrix

$x_B \geq 0$

$x_n = 0$

- Basic feasible points are corner points

If there is feasible point, then there is a basic feasible point

If an optimal sol exist, then at least 1 basic feasible point is opt sol.

- LP is convex opt. problem

thus KKT are sufficient conditions

+) Algorithm : Simplex Method

Loop
- If $\lambda_{I,N} > 0$ terminate
- Choose index q : $q \notin T^k(x)$, $\lambda_{I,q} = \min_{i \notin T^k(x)} \lambda_{I,i}$ note $\lambda_{I,q} < 0$
- Init $x_q^+ = 0$, fix all $\underline{x}_N^+ = \underline{0}$
- Increase x_q^+ following $Ax^+ = b$ until some x_p^+ with $p \in T(x)$ becomes 0

(Include index q of the inequal. const. that failed; $\lambda_{I,q} < 0$) that the most negative variable (not the basic)

Remove: the variables which in that row, limit the most the changes of the new variable

Strong duality + Primal, dual feasibility $\Big\}$ $\Rightarrow$ necessary & sufficient conditions for LP

+) Duality: at the Optimum

Primal problem: $\min_x d^T x$ s.t. $Ax = b$, $x \geq 0$

Dual prob.: $\min_{\lambda_E} -b^T \lambda_E$ s.t. $d - A^T \lambda_E \geq 0$, ~~$\lambda_E \geq 0$~~

~~Choose~~ Let Langrange multiplier as λ_E $\qquad x$

$\Rightarrow$ Langrange func: $L(x, \lambda_E, \lambda_I) = d^T x + \lambda_E^T (b - Ax) - \lambda_I^T x$

$\bar{L}(\lambda_E, x) = -b^T \lambda_E + x^T (A^T \lambda_E - d)$

$\Rightarrow$ KKT conditions:

$A^T \lambda_E^* + \lambda_I^* = d$

$Ax^* = b$

$x^* \geq 0$

$\lambda_I^* \geq 0$

$\lambda_{I,i}^* \cdot x_i^* = 0 \quad \forall i = 1, \dots, n$

$\Leftrightarrow$

$d^T x^* = (A^T \lambda_E^* + \lambda_I^*)^T x^*$

$= \lambda_E^{*T} A x^* + \lambda_I^{*T} x^*$

$= \lambda_E^{*T} \cdot b$

$= b^T \cdot \lambda_E^*$

If choose $\lambda_{I,i}^* = d_i - a_i^T \lambda_E^*$

$Ax^* = b$

$A^T \lambda_E^* \leq d \Leftrightarrow \lambda_I^* \geq 0$

$x^* \geq 0$

$x_i^* (a_i^T \lambda_E^* - d_i) = 0 \quad \forall i$

$\Leftrightarrow x_i^* \lambda_{I,i}^* = 0 \quad \forall i$

$\Rightarrow$ Strong duality: $\boxed{d^T x^* = b^T \lambda_E^*}$ (true for all convex problem)

L7

+) <u>Primal - Dual Method (PDM)</u>

$$\min_x \; d^T x$$
$$\text{st. } Ax = b$$
$$-x_i \leq 0$$

$$\Rightarrow L = d^T x + \lambda_E^T (Ax+b) - \lambda_I^T x$$

KKT conditions:

$$\begin{cases} (1) & A^T \lambda_E + \lambda_I = d \\ (2) & Ax = b \\ (3) & x_i \lambda_{I,i} = 0, \quad i = 1, 2, \dots n \\ (4) & x \geq 0, \quad \lambda_I \geq 0 \end{cases}$$

$$\Rightarrow F(x, \lambda_E, \lambda_I) = \begin{bmatrix} A^T \lambda_E + \lambda_I - d \\ Ax - b \\ X \Lambda_I e \end{bmatrix} = 0$$

$$X = \operatorname{diag}(x_1 \dots x_n); \quad \Lambda_I = \operatorname{diag}(\lambda_{I,1} \dots \lambda_{I,n}), \quad e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Adapted Newton method

$$\begin{cases} A^T \lambda_E + \lambda_I = d \\ Ax = b \\ \underline{x_i \lambda_{I,i} = \tau} \quad i = 1, 2, \dots n \qquad \tau > 0 \\ x_\tau > 0, \; \lambda_{I,\tau} > 0 \end{cases}$$

μ: average violation of nonlinear constraints

$\tau = \sigma \mu; \quad \sigma \in [0, 1]$

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \lambda_{I,i} = \frac{x^T \lambda_I}{n}$$

$\Rightarrow$ <u>Algorithm</u>:

+) Initial feasible guess $x_0 \geq 0, \; \lambda_E^0, \; \lambda_I^0 > 0$

+) Loop k: solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \Lambda_I & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \delta x \\ \delta \lambda_E \\ \delta \lambda_I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X \Lambda_I e + \sigma \mu e \end{bmatrix}; \quad \sigma^{(k)} \in [0, 1], \quad \mu^{(k)} = \frac{x^{(k)T} \lambda_I^{(k)}}{n}$$

Set $\left[x^{(k+1)}, \lambda_E^{(k+1)}, \lambda_I^{(k+1)} \right] = \left[x^{(k)}, \lambda_E^{(k)}, \lambda_I^{(k)} \right] + \alpha_k \left[\delta x, \delta \lambda_E, \delta \lambda_I \right]$

Find α_k such $x^{(k+1)} > 0, \; \lambda_I^{(k+1)} > 0$

T) Quadratic Programming:

$$\min_{x} \ \frac{1}{2} x^T G x + d^T x \qquad G_{n\times n}$$

$$\text{s.t.} \quad a_i^T x - b_i = 0, \ i \in \mathcal{E}$$

$$a_i^T x - b_i \leq 0 \ ; \ i \in \mathcal{I}$$

If $G \succeq 0$ semi definite $\Rightarrow$ QP is convex

G is indefinite $\Rightarrow$ QP not convex

QCQPs have both quadratic obj func & quadratic constraints

$\Rightarrow$ KKT:
$$\begin{cases} Gx^* + d^T + A^T \lambda^* = 0 \\ a_i^T x - b_i = 0 \quad \forall i \in \mathcal{E} \\ a_i^T x - b_i \leq 0 \quad \forall i \in \mathcal{I} \\ \lambda_i^* \geq 0 \quad \forall i \in \mathcal{I} \\ \lambda_{\mathcal{I}}^* (a_i^T x - b_i) = 0 \quad \forall i \in \mathcal{I} \qquad \text{bilinear} \end{cases}$$

ANU

Constrained Problem

L8)

1) Elimination of variables: - Reduced space formulation

$$\min_{x \in \mathbb{R}^n} f(x)$$
s.t. $c_i(x) = 0, \ i \in E$

$$x = \begin{bmatrix} y \\ z \end{bmatrix} \begin{matrix} n-m \\ m \end{matrix} \Rightarrow \min_y \tilde{f}(y)$$

- Need to be able solve $c_i(x)$ for $z(y)$
- Possible to linear, some nonlinear

+) Elements to consider when choosing solver:

Arithmetic complexity: CPU time = # iterations * CPU time / iteration

+) Performance plot

2) Quadratic penalty method (QPM)

+) $\min_{x \in \mathbb{R}^n} f(x)$ s.t. $c_i(x) = 0$, $i \in \mathbb{E}$

$$\Rightarrow Q(x;\mu) = f(x) + \frac{1}{2\mu} \sum_{i \in \mathbb{E}} \left[c_i(x)^2\right] \qquad \mu > 0$$

Construct sequence $\{\mu^{(k)}\}$ with $\lim_{k \to \infty} \mu^{(k)} = 0$

+) $\min_{x \in \mathbb{R}^n} f(x)$ s.t. $c_i(x) = 0$ $i \in \mathbb{E}$
  $c_i(x) \leq 0$ $i \in \mathbb{I}$

$$\Rightarrow \boxed{Q(x;\mu) = f(x) + \frac{1}{2\mu} \sum_{i \in \mathbb{E}} [c_i(x)]^2 + \frac{1}{2\mu} \sum_{i \in \mathbb{I}} \max[0, c_i(x)]^2}$$

+) Algorithm:
- Given $\mu^{(1)} > 0$, $\varepsilon^{(1)} > 0$, $x^{(0)}$
- Step k: solve $x_k = \operatorname{argmin} Q(x, \mu^{(k)})$
  then reduce $\mu^{(k)}$
- Stop when constraint violation are sufficiently small

$\lim_{\mu \to 0} \hat{p}^*(\mu) = p^*$

μ should be small

+) For equality constraints, penalty function is smooth
  inequality ———————————— nonsmooth

3) Augmented Lagrangian Method (ALM)

$\min_{x \in \mathbb{R}^n} f(x)$ s.t. $c_i(x) = 0$ $i \in \mathbb{E}$

$$\Rightarrow L_A(x;\lambda;\mu) = f(x) + \sum_{i \in \mathbb{E}} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i \in \mathbb{E}} [c_i(x)]^2$$

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} + \frac{c_i(x^{(k)})}{\mu^{(k)}}$$

fewer iterations even for large $\mu^{(k)}$
better conditioning

4) Log-Barrier Method — for inequalities ⇒ interior point; equalities ⇒ not interior

+) $\min_{x \in \mathbb{R}^n} f(x)$ s.t. $c_i(x) \leq 0 \quad i \in \mathcal{I}$

$$\Rightarrow P(x,\mu) = f(x) - \mu \sum_{i \in \mathcal{I}} \log[-c_i(x)] \qquad \mu > 0$$

+) for $c_i(x) = 0 \quad i \in \mathcal{E}$ & $c_i(x) \leq 0 \quad i \in \mathcal{I}$

$$\Rightarrow B(x,\mu) = f(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} [c_i(x)]^2 - \mu \sum_{i \in \mathcal{I}} \log[-c_i(x)]$$

5) Linearly Constrained Lagrangian Method (LCL)

Linearize the constraints

+) $\min_{x \in \mathbb{R}^n} f(x)$ s.t. $c_i(x) = 0 \quad i \in \mathcal{E}$

$$\Rightarrow \min_{x \in \mathbb{R}^n} F^{(k)}(x) \quad \text{s.t. } \nabla c_i (x^{(k)})^T (x - x^{(k)}) + c_i(x^{(k)}) = 0 \quad i \in \mathcal{E}$$

$$F^{(k)}(x) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i^{(k)} \bar{c}_i^{(k)}(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} [\bar{c}_i^{(k)}(x)]^2$$

$$\bar{c}_i^{(k)}(x) = c_i(x) - c_i(x^{(k)}) - \nabla c_i (x^{(k)})^T (x - x^{(k)})$$

6) SQP Sequential Quadratic Programming

ANO

Integer Optimization

L9,

1, Introduction:

+, Most general formulation:

$$\min_{\underline{x}, y} f(\underline{x}, y)$$
$$\text{s.t.} \quad c_i(\underline{x}, y) = 0, \ \forall i \in E$$
$$c_i(\underline{x}, y) \leq 0, \ \forall i \in I$$

$\underline{x} \in \mathbb{R}^{n_x}$ continuous variables

$y \in Y$ discrete variables

(e.g. $y \in \{0,1\}^{n_y}$)

+, Semi general formulation

$$\min_{\underline{x}, y} f(\underline{x}) + \underline{d}^T y$$
$$\text{s.t.} \quad c_i(\underline{x}) = 0 \ \ \forall i \in E$$
$$c_i(\underline{x}) + \underline{a}_{y,i}^T y \leq 0 \ \ i \in I$$

+, Formulation matters

2, Theory ?

+, Convexity: non convex (obviously)

$\Rightarrow$ misnomer: when call sth convex mixed-integer program

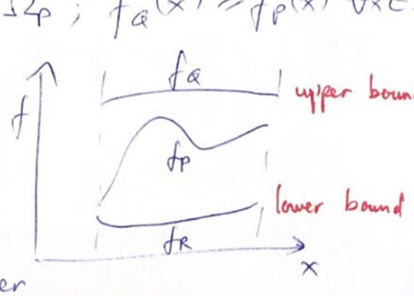
+, Restrictions, Relaxations & Approximations

- Problem P, feasible set Ω, objective func f.
- R is a relaxation of P if $\Omega_R \supset \Omega_P$; $f_R(x) \leq f_P(x) \ \forall x \in \Omega_P$
- Q is a restriction of P if $\Omega_Q \subset \Omega_P$; $f_Q(x) \geq f_P(x) \ \forall x \in \Omega_P$

e.g. $y_k \in \{0,1\}$

relaxed to $y_k \in [0,1]$

restricted to $y_k = 0$



- Approximation could be relaxation, restriction or neither

3) Branch & bound algorithm

for all func { linear / convex nonlinear / nonconvex nonlinear

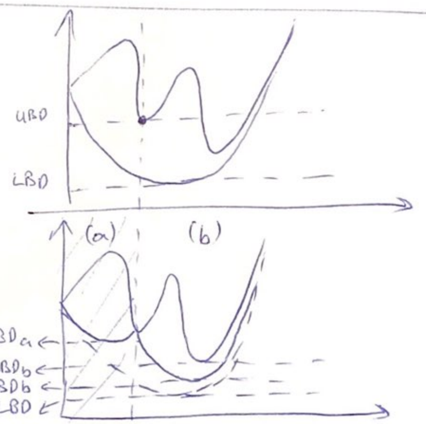
- Construct a series of restricted MILPs and solve their LP relaxation
- Branching step: choose $y_j \in \{0, 1\} \Rightarrow$ 2 sub-prob MILP
  children inherit lower bound from parent
- Node selection: heuristic (best lower bound, breadth-first, depth-first ..)
- If the node is not feasible (not integer)
  $\Rightarrow$ expand it, use the current value as lbd
- Optimality gap $G = f^{ub} - f^{lb}$

  Exact termination for $G = 0$

  typically approximation: terminate if $G \leq a_{tol}$ or $G \leq r_{tol} \cdot (|f^{ub}| + \varepsilon)$

  a_{tol}, r_{tol}: user-specified $\qquad \varepsilon = 10^{-10}$

L10) 4) Branch and Bound for Box-constrained NLPs



1. Construct a relaxation
2. Solve relaxation $\Rightarrow$ LBD
3. Solve original locally $\Rightarrow$ UBD
4. Branch to nodes (a) and (b)
5. Repeat till convergence / or termination condition

non convex means we will not only have global solution, but also local ones

5) Convex relaxation of non-convex functions

+) Natural Interval Extension:

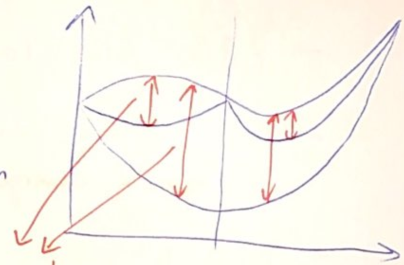
- Decompose function to finite sequence of simpler funcs
- Keep propagate intervals

$\Rightarrow$ Simple, cheap $\Rightarrow$ but weak

$\Rightarrow$ Improvement with Centered form / Taylor models

+) **αBB Method** add negative quadratic term

- $f(x) + \sum_i \alpha_i (x_i - x_i^L)(x_i - x_i^U)$
  - is relaxation for any $\alpha > 0$
  - convex for large enough α
  - suitable α from underestimating eigenvalues of Hessian



- As $x_i^U - x_i^L \to 0 \Rightarrow$ converge

  the smaller the range, the stronger / tighter the relaxation

  assume smooth func

+) **Multivariate McCormick** in original variable space

convexify → linearize

+) **Auxiliary variable method** in higher dimension space

- replace with new variables
- convexify
- linearize

6, Convergence rate of convex relaxation

$x \in X^0 = [x^{L,0}, x^{U,0}] \supset X = [x^L, x^U]$

$\delta(x) = \max_i \{x_i^U - x_i^L\}$

- Construct
  - convex f^u such $f^u(x) \le f(x) \le f^o(x) \quad \forall x \in [x^L, x^U]$
  - concave f^o

$\Rightarrow$ Convergence: $f^o(x), f^u(x) \to f(x)$ for $\delta(X) \to 0$

+) Pointwise convergence rate: γ

$\exists C > 0$, s.t. $\forall X \in X^0$: $\sup_{x \in X} \{f(x) - f^u(x), f^o(x) - f(x)\} \le C(\delta(X))^\gamma$

+) Hausdorf convergence rate β $\exists C > 0$, s.t. $\forall X \subset X^0$:

$\max\{\inf_{x\in X} f(x) - \inf_{x \in X} f^u(x), \sup_{x\in X} f^o(x) - \sup_{x\in X} f(x)\} \le C(\delta(X))^\beta$

+) Cluster effect: need high convergence rate, avoid too many nodes..

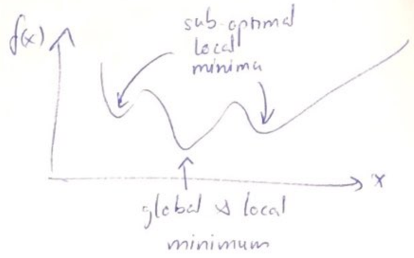
desired > 2

problematic < 2 $\qquad \gamma \le \beta$

acceptable $= 2$

7) Deterministic global solvers

(when we have non-convex prob)



- LBD from local methods
- UBD from any feasible points (typically converges faster than LBD)
- Need to prove convergence

⊛ For MINLP:

typically do B&B on integer variable y, then globally solve each NLP at each node

8) Reduced Space for Global Optimization:

Full-space

$$\min_{x,z} f(x,z)$$
$$\text{s.t.: } c_I(x,z) \leq 0$$
$$c_E(x,z) = 0$$

with $\dim(x) \ll \dim(z)$

→ Solve equa. contstraints $c_E(x,z) = 0$ for z →

Reduce space

$$\min_{x} \tilde{f}(x)$$
$$\text{s.t. } \tilde{c}_I(x) \leq 0$$

Total $\dim(x)$

x: dof

z: state variables

Smaller problem ⇒ easier, faster

9) Stochastic global optimization:

- Black box optimization
- General idea: sample the space → avoid getting trapped with local suboptimal
  - (+) robust, no derivatives required, easy ..
  - (−) slower than grad-based ..
    no guarantee optimality

+) No free-lunch Theorem

convex

+) Random search

- Start from initial point $x^{(0)}$
- Randomly choose new iterate $x^{(k+1)}$
- Compare $f(x^{(k+1)})$ and best value found f^*

+) Multistart:

- Start local solvers with many initial guesses
- Pick initial guesses with grid, latin hypercube, random..
- Try Parallelize..

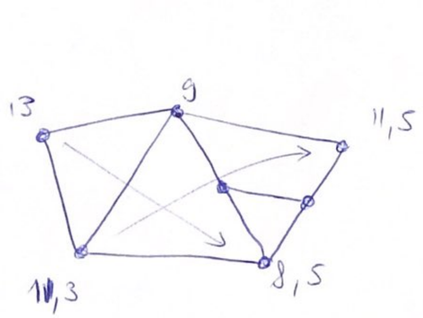
+) Genetic Algorithm

[Mutation: peturb entries randomly
Crossing: (recombine) child inherits some entries from parents

16) Derivative free optimization: only numerical evaluation

- when gradient maybe expensive or unavailable ..
- Can deal with non-smoothness

+) Simplex search



+) Univariate Search

+) Pattern Search

ANO

Dynamic Optimization

L11,

1, Dynamic Optimization : Examples and Solution Strategies

+) General formulation:

$$\min_{x(\cdot),\, u(\cdot)} \Phi(x(t_f)) \quad \text{s.t.} \quad \begin{array}{ll} \dot{x}(t) = f(x(t), u(t)),\ t \in [t_0, t_f] & \text{state equations} \\ x(t_0) = x_0 & \text{initial conditions} \\ g_P(x(t), u(t)) \leq 0 \quad \forall t \in [t_0, t_f] & \text{path constraints.} \\ g_T(x(t_f)) \leq 0 & \text{terminal conditions} \end{array}$$

$u(t) \in \mathbb{R}^m$ controls
$x(t) \in \mathbb{R}^n$ states

+) Note: - x_0 can also be additional optimization variables
- Final time t_f assumed fixed, can be reformulated by scaling
- Φ of Mayer type, integral obj. func. can be reformulated using additional state

depends only on final time

Dynamic Optimization Problem
- Direct Methods
  - Simultaneous Method
  - Single Shooting
  - Multiple Shooting
- Indirect Methods
  - Adjoint Equations

2, Simultaneous Method : – Full discretization

Mostly used now, due to continuity of research and development of solvers

+) Discretization scheme:

- Explicit Euler $c_k(y) = x_{k+1} - x_k - \tau f(x_k, u_k) = 0$ — bad accuracy, sth instability
- Implicit Euler $c_k(y) = x_{k+1} - x_k - \tau f(x_{k+1}, u_{k+1}) = 0$ — pose no additional complication

+) Discretize : obj func: $\Phi(x_M) = \hat{F}(y)$

⇒ get a NLP with → new variable: $y^T = [u_0; x_0, u_1, x_1 \dots, u_{M-1}, x_{M-1}]$

transition func: $\dot{x}(t_k) \approx \frac{1}{\tau}(x_{k+1} - x_k) \Rightarrow c_k(y) = x_k + \tau f(x_k, u_k) - x_{k+1} = 0$

+) Large but sparse NLP: $A = \nabla_y c(y)$

+) Method : ⇒ interior point
active set has combinatorial complexity

3, Collocation: as a better discretization scheme

+) General idea: Use s^{th} degree polynomial $P_s(t)$

+) Formulation

4, Single Shooting – Sequential Method – Late Discretization

12)

Single shooting is one of Sequential Methods

- Key idea: with Simultaneous method, we discretize both states x and control u

However, the real dof is just the control u

⇒ We will discretize only the control variables with this method

- Approximate $u(t) \in \mathbb{R}^{n_u}$ by series of (orthornormal) basis func $\sigma_{i,k}(t)$

$$u_i(t) \approx \sum_{k=1}^{s} \underbrace{p_{i,k}}_{\text{coefficient}} \underbrace{\sigma_{i,k}(t)}_{\text{basis func}} \quad , \; i = 1, 2, \ldots n_u$$

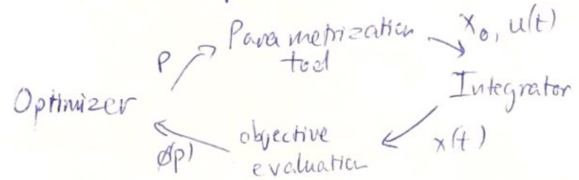
parametrization

$$\Rightarrow \min_{x(\cdot),\, u(\cdot)} \Phi(x(t_f)) \qquad \Longrightarrow \qquad \min \phi(p)$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t)) \qquad\qquad \text{s.t. } c(p) \leq 0$$

$$x(t_0) = x_0$$

$$u^L \leq u(t) \leq u^u$$



- But need (at least) 1st-order derivatives $\nabla\phi(p)$ cause with $\left.\begin{array}{l}\dot{x}(t) = f(x(t), u(t)) \\ x(t_0) = x_0\end{array}\right\} \Rightarrow$ Can evaluate $\phi = \Phi(x(t_f))$

(*) Draw the graph

+) To calculate Gradients

$$\nabla\phi(p) = \left[\left.\frac{\partial \phi}{\partial p_1}\right|_p \quad \cdots \quad \left.\frac{\partial \phi}{\partial p_j}\right|_p \right]^T$$

① Gradients by Perturbation:

forward finite differences

$$\left.\frac{\partial \phi}{\partial p_j}\right|_p = \frac{\phi(p + e_j \Delta) - \phi(p)}{\Delta} \qquad e_j = [0, \ldots, 1, 0 \ldots]^T$$

(+) easy to implement

(-) limited approximation quality; high computational cost

② Gradients by Forward Sensitivities:

- Chain rule: $\left.\frac{\partial \phi}{\partial p_j}\right|_p = \sum_i \left(\left.\frac{\partial \Phi}{\partial x_i}\right|_p \cdot \underbrace{\left.\frac{\partial x_i}{\partial p_j}\right|_{p,\, t = t_f}}_{= S_{i,j} \text{ (Sensitivities)}} \right)$

- $\dot{x}(t) = f(x(t), u(t))$

$$\Rightarrow \left.\frac{d\dot{x}}{dp}\right|_t = \left.\frac{df}{dp}\right|_{x(t),\, p,\, u(t)} = \underbrace{\left.\frac{\partial f}{\partial x}\right|_{x(t),\, p,\, u(t)}}_{\text{(from } f\text{)}} \cdot \left.\frac{dx}{dp}\right|_t + \underbrace{\left.\frac{\partial f}{\partial u}\right|_{p,\, u(t)}}_{\text{(from } f\text{)}} \cdot \underbrace{\left.\frac{du}{dp}\right|_{p,t}}_{\text{(from parametrization)}}$$

$$\Rightarrow \dot{S}(t) = \left.\frac{d\dot{x}}{dp}\right|_t = \left.\frac{\partial f}{\partial x}\right|_{x(t),\, p,\, u(t)} \cdot S(t) + \ldots$$

t

(+) { exact in theory, accurate in practice
if implemented efficiently, faster than finite differences

(-) { hard to implement efficiently
computational effort scales with # params

③ Adjoint (backward mode)

(+) computational effort doesn't scales with #inputs

(-) { ——————— scales with #outputs
very hard to understand & implement efficiently

④ Advantages & disadvantages of Sequential Method

(+) enable error-controlled integration
applicable to large, sophisticate models
suitable for black box algorithms

(-) problematic for unstable system
2nd derivatives are expensive

Choose solver (NLP solver) requires few (major) iterations

| while for simultaneous method, choose solver that scales well with many variables & constraints

⇒ 5) Multiple shooting

as a method to compensate for the disadvantages of Single Shooting (when we have unstable systems)

Add addition state value as initial point

→ { split the intervals
gain additional dof

→ { sensitivity decrease ↓
but problem size increase ↑

+) Indirect Methods

- Try to set up optimality conditions then solve them by a discretization method
- Difficult or impossible for problems with path constraints

+) Software tools for Dynamic Optimization

6) Path constraints & switching structure

$$\min_{x(\cdot),\, u(\cdot)} \phi(x(t_f))$$

s.t. $\dot{x}(t) = f(x(t), u(t))$

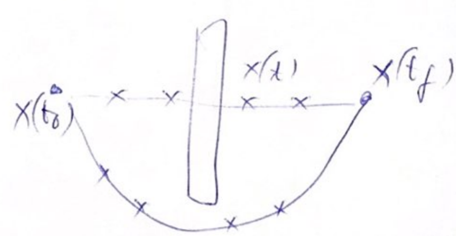
$x(t_0) = x_0$

$g_p(x(t), u(t)) \leq 0, \quad t \in [t_0, t_f]$ path constraints

- Typically enforced on finite # times

$g_p(x(t_k, p), u(t_k, p)) \leq 0, \quad k = 0, \dots N$

- Violations between points possible



+) SIP

+) Switching structure:

7) NMPC (Nonlinear Model Predictive Control)

When we do Dynamic Optimization, we need to worry about CPUs time, at least when we want to take it online

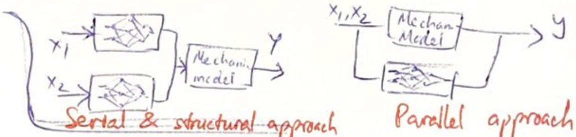
Machine Learning & Optimization

43) 1) Machine Learning & Hybrid modelling

- Machine Learning: programming computers + data/experience
- Interpolation: prediction within the convex hull of training data $\gtrless$ extrapolation

⇒ Data-driven models can't extrapolate

- Hybrid Models: combine Mechanistic & Data-driven Models



2) Artificial Neural Networks

- ANNs: adopt idea of neurons ⇒ nodes, activation func, layers...
- Deep learning: branch of ML, using deep ANN

3) Training of data-driven models: ANN

- Back-propagation: gradients go back layer by layer
- Divide data:
  - train set 70%
  - validation set 15%
  - test set 15%
- Regularization:
  - drop out
  - weight decay (penalize large weight $E = L + \sum w^2$

4) ~~Determination~~ Deterministic global optimization with ANN embedded

Full space

$$\min_{x,z} f(x,z)$$
$$\text{s.t. } g(x,z) \le 0$$
$$h(x,z) = 0$$

$\dim(x) \ll \dim(z)$

Solve $h(x,z) = 0$ for z

Reduced space

$$\min_{x} \tilde{f}(x)$$
$$\text{s.t. } \tilde{g}(x) \le 0$$

save computational time

5) Gaussian Processes (Kriging) & Bayesian Optimization

+) Gaussian Processes (a.k.a. Kriging)

- A stochastic process
- Generalization of multivariate Gaussian distributions

+) Bayesian Optimization: sequential design strategy for global optimization problem of black-box functions

-

- Gaussian Processes lead to large-scale nonlinear problem in the full-space formulation

⇒ The reduced-space formulation reduces the problem size drastically

L14) Optimization under Uncertainty & SIP

1) Parametric Optimization:

$$\min_x f(x,y) \qquad \text{(PAR)}$$
$$\text{s.t. } c(x,y) \leq 0$$

y as $\left[\begin{array}{l}\text{uncertain model params} \\ \text{stochastic process}\end{array}\right.$

- $\left.\begin{array}{l}\text{optimal solution} \\ \text{obj. value}\end{array}\right\}$ depend on params

+) Parametric optimization is only useful when uncertainty is realized before the decision variables must be fixed

+) Parameter regions: in which, solutions stay qualitatively the same

2) Introduction to optimization under uncertainty:

+) Stochastic approaches: – consider probability measures over possible uncertainty

⇒ – known probability for y
– objective $\min_{x} \mathbb{E}_y [f(x,y)]$
– chance for feasibility

+) Robust approaches: consider worst case

– y bounded, $y \in Y$
– $\min_{x} \max_{y \in Y} f(x,y)$
– Guaranteed feasibility

Both: more $\left[\begin{array}{l}\text{reliable} \\ \text{conservative}\end{array}\right.$ solutions

3) Two-Stage Stochastic Optimization:

$$f(x,y,z) = f^{u}(x) + f^{L}(y,z)$$

x – decision variable must be fixed before uncertainty is realized

z – ______ can be after ______

$$\Rightarrow \min_{x} f^{u}(x) + \mathbb{E}_y (F(x,y))$$
$$\text{s.t. } c^{u}(x) \leq 0$$
(ST1)

$$F(x,y) = \min_{z} f^{L}(y,z)$$
$$\text{s.t. } c^{L}(x,y,z) \leq 0$$
(ST2) a specific uncertainty scenario

+) A Single stage formulation:
– y has continuous or large discrete distribution
– $\mathbb{E}_y(F(x,y))$ is difficult / if not impossible to evaluate
– Can approximate by finitely many samples

$$\min_{x, z_s} f^{u}(x) + \sum_{s \in S} P_s f^{L}(y_s, z_s)$$
$$\text{s.t. } c^{u}(x) \leq 0$$
$$c^{L}(x, y_s, z_s) \leq 0 \ \forall s \in S$$

$\mathcal{Z}$

4) Introduction to Semi Infinite Programs

+) $\min_{x \in X} f(x)$

s.t. $c(x,y) \le 0, \ \forall y \in Y$

As the cardinality of $|Y| = \infty$, we have a SIP

(there is infinite number of elements in set Y)

$\Rightarrow$ We have finitely many variables

but infinitely many constraints

+) Some application:
- Design Centering (cutting diamond)
- Path constraints in Dynamics Optimization

+) If $\exists \max c(x,y)$, then we can reformulate as as LLP (Low level problem)

$$\min_{x \in X} f(x)$$
$$\text{s.t. } 0 \ge \max_{y \in Y} c(x,y)$$

5) Basic solution methods for SIP

- Intuitive approach may not be the best (KKT conditions) not sufficient

- Local reduction
- Discretization: replace Y by a finite discretization Y^D
  $\rightarrow$ a finite approximation of SIP (a relaxation)

- Blankenship & Falk: $Y^{LBP} \subset Y$, $|Y^{LBD}| < \infty$

$$\max_{y \in Y} c(x,y) < \max_{y \in Y^{LBD}} c(x,y)$$

- UBD problem: $c(x,y) \le -\varepsilon \quad \forall y \in Y^{UBD}$

Unconstrained problem

Iterative Descent

Solve Optimal Conditions

Line Search

Trust Region Approach

$x_{k+1} = x_k + p_k$; $m_k = g_k^T p_k + \frac{1}{2} p_k^T H_k p_k$ then SQP, Lagrange

$p_k = \arg\min_{p_k \in \Delta} m_k(p)$

$\rho_k = \frac{f(x^{(k)}) - f(x^{(k)} + p^{(k)})}{m^{(k)}(0) - m^{(k)}(p^{(k)})}$

Step length

- Armijo condition
- Improved Line search
- Wolfe Condition
- Golden section search

Descent Direction

- Steepest Descent $x^{(k+1)} = x^{(k)} - \alpha_k \cdot \nabla f(x^{(k)})$ $\quad O\left(\frac{1}{\varepsilon^2}\right)$
- Newton - Raphson Method $p^{(k)} = -\left[\nabla^2 f(x^{(k)})\right]^{-1} \nabla f(x^{(k)})$ $\quad O\left(\frac{1}{\varepsilon^{2-\tau}}\right)$ $\quad 1 > \tau > 0$
- Inexact Newton Method $H^{(k)} \cdot p^{(k)} = -g^{(k)}$ solve approximately
- quasi Newton — approx. $H^{(k)-1}$
- Modified Newton Method — $H^{(k)}$
- Gauss - Newton Method $f(x) = \min_{x \in \mathbb{R}^e} \frac{1}{2} e^T(x) \cdot e(x)$

Constrained Problem

Transform to unconstrained problem

Elimination of variables
solve $c_i(x) = 0$

only equality constraints

Barrier func. (Log-Barrier)
$$B(x,\mu) = f(x) + \frac{1}{2\mu}\sum_{i\in E}[c_i(x)]^2 - \mu \sum_{i\in I}\log[-c_i(x)]$$

$B(x,\mu)\Uparrow$ as x close to edge of P

Penalty func (Quadratic penalty method) QPM
$$Q(x,\mu) = f(x) + \frac{\lambda}{2}\sum_{i\in E}[c_i(x)]^2 + \frac{\lambda}{2}\sum_{i\in I}\max[0, c_i(x)]^2$$

$\lambda \to \infty \Rightarrow \hat{p}^* \to p^*$

$$\begin{cases} P(p) = 0 & p \in P \\ P(p) > 0 & p \notin P \end{cases}$$

Augmented Lagrangian Method (ALM)
$$L_A(x,\lambda,\mu) = f(x) + \lambda^T \sum_{i\in E} c(x) + \frac{1}{2\mu}\sum_{i\in E} c^T(x).c(x)$$

Simplify cases

Linear Programming
- Simplex Method
- Primal Dual Method (PDM)

$$\min_x \; c^T x$$
s.t. $a_i^T x - b_i = 0$

$-x \leq 0$

Interior Points
- Use Barrier functions

~~then Lagrange SQP~~

then Newton-Raphson (for convex opt. prob.)

- Or Lagrange func

Quadratic Programming
$$\min \frac{1}{2}x^T G x + d^T x$$
s.t. $a_i^T x - b_i = 0, \quad i \in E$

$a_i^T x - b_i \leq 0, \quad i \in I$

Linearly Constrained Lagrange Method (LCL)
linearize constraints

then run ALM

SQP
transform to QP by approximation (obj func and constraints)

Newton method

transform to QP by approximation (Lagrange func & constraints)

KKT Conditions

ANM?

Convex relaxation: of nonconvex - functions

	Natural interval extensions	αBB	McCormick (original & Multivariate)
Point wise γ		2 even for fixed α	2 mild assumptions
Hausdorf β	1	(+) only smooth func	

$\gamma \leq \beta$

Dynamics optimization

1) Full-discretization - Simultaneous Method

- Main drawback: quite large, new variable size
  thus also ~~many~~ more constraints

2) Single Shooting:

- Gradients:
  - by Pertubation (+) easy
    (-) not so accurate
    computational cost
  - by Forward Sensitivities

- Accurate

  Hard

  Problem with unstable system

  Need solver scales ~~well~~ requires few major iterations

+) Stochastic global optimization

- Random Search
- Multistart
- Genetic Algorithm

+) Derivative - free ~~algo~~ optimization

- Simplex search
- Univariate search
- Pattern search

Exercise

~~T1~~

- 0, Introduction to MATLAB
- 1, Formulation of optimization problem
- 2, Grad, Hessian, Optimality conditions
- 3, Grad, Steepest Descent
- 4, Optimality Conditions
- 5, KKT conditions
- 6, Simplex Method
- 7, Interior Point Method
- 8, Penalty & SQP Method
- 9, Branch & Bound Method for MILPs
- 10, Branch & Bound Method for MIPs
- 11, Full discretization approach
- 12, Single Shooting
- 13, Sensitivity Equations