02610
Optimization and Data Fitting
Week 12: Introduction to Constrained Optimization

Yiqiu Dong

DTU Compute
Technical University of Denmark

Constrained optimization problems

$$
\begin{aligned}
\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) \\
\text{subject to} \quad & c_i(\boldsymbol{x}) = 0 \text{ for } i \in \mathcal{E} \\
& c_i(\boldsymbol{x}) \geq 0 \text{ for } i \in \mathcal{I}
\end{aligned}
$$

- $\boldsymbol{x}$ is the unknown vector.
- f is the objective function.
- c_i are constraint functions, and $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints, respectively.
- **Assumption:** f and c_i are all smooth.
- Feasible set: The set of all possible $\boldsymbol{x}$, i.e., the points satisfy all constraints.

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : c_i(\boldsymbol{x}) = 0, i \in \mathcal{E}; c_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}.$$

What is a solution?

$$
\begin{aligned}
\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) \\
\text{subject to} \quad & c_i(\boldsymbol{x}) = 0 \text{ for } i \in \mathcal{E} \\
& c_i(\boldsymbol{x}) \geq 0 \text{ for } i \in \mathcal{I}
\end{aligned}
$$

We would most like to have global minimizer, i.e.,

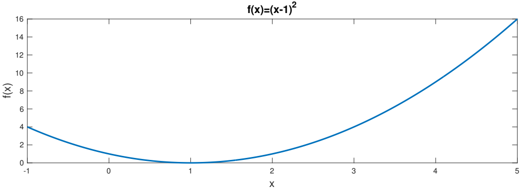
A points $\boldsymbol{x}^*$ is a global minimizer if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega$ with $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : c_i(\boldsymbol{x}) = 0, i \in \mathcal{E}; c_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}$.

But the global minimizer can be difficult to find due to limited knowledge of f, most algorithms are able to find only a local minimizer, i.e.,

A points $\boldsymbol{x}^*$ is a local minimizer if $\boldsymbol{x}^* \in \Omega$ and there is a neighborhood $\mathcal{N}$ of $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{N} \cap \Omega$.

Example 1

$$\min_{x \in \mathbb{R}} f(x) = (x-1)^2$$



- With the constraint $x \geq 0$, the minimizer is $x^* = 1$.
- With the constraint $x - 2 \geq 0$, we have $\Omega = [2, +\infty)$ and the minimizer is $x^* = 2$.
- With the constraint $x - 2 \geq 0$ and $3 - x \geq 0$, we have $\Omega = [2, 3]$ and the minimizer is $x^* = 2$.
- With the constraint $3 - x = 0$, we have $\Omega = \{3\}$ and the minimizer is $x^* = 3$.
- With the constraint $3 - x \geq 0$ and $x - 4 \geq 0$, we have $\Omega = \emptyset$ and the minimizer does not exist.

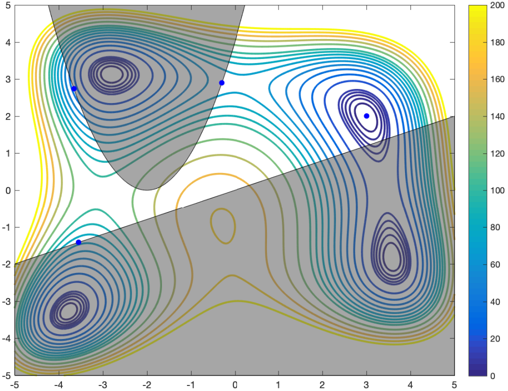
Example 2

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \quad f(\boldsymbol{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)$$

$$\text{subject to} \quad c_1(\boldsymbol{x}) = (x_1+2)^2 - x_2 \geq 0$$

$$c_2(\boldsymbol{x}) = -4x_1 + 10x_2 \geq 0$$

Feasible set (not grey)



Blue dots:
local minimizers
(total 4)

Active set

At a feasible point $\boldsymbol{x}$, the inequality constraint c_i $(i \in \mathcal{I})$ is:

- *active* iff $c_i(\boldsymbol{x}) = 0$ ($\boldsymbol{x}$ is on the boundary for that constraint);
- *inactive* iff $c_i(\boldsymbol{x}) > 0$ ($\boldsymbol{x}$ is interior point for that constraint).

The active set $\mathcal{A}(\boldsymbol{x})$ is the set of indices of equality constraints and active inequality constriants:

$$\mathcal{A}(\boldsymbol{x}) = \mathcal{E} \cup \{i \in \mathcal{I} : c_i(\boldsymbol{x}) = 0\}.$$

- An inequality constraint which is inactive at $\boldsymbol{x}$ has no influence on the optimization problem in a neighborhood of $\boldsymbol{x}$.

Convex optimization problems

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & c_i(\boldsymbol{x}) = 0 \text{ for } i \in \mathcal{E} \\ & c_i(\boldsymbol{x}) \geq 0 \text{ for } i \in \mathcal{I} \end{aligned}$$

- The feasible set Ω must be convex.
- The objective function f must be convex on Ω.
- The equality constraint functions c_i with $i \in \mathcal{E}$ must be linear.
- The inequality constraint functions c_i with $i \in \mathcal{I}$ must be concave.

Properties

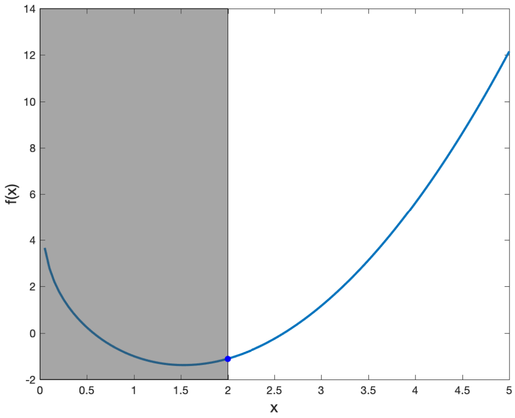
If Ω is bounded and convex and if f is convex on Ω, then

- any local minimizer $\boldsymbol{x}^* \in \Omega$ is a global minimizer.
- If f is strictly convex, then the global minimizer is unique.

Example: Convex problem 1

$$\min_{x} \; f(x) = (x-1)^2 - \sqrt{x} - \ln(x)$$
$$\text{s.t. } c(x) = x - 2 \geq 0$$

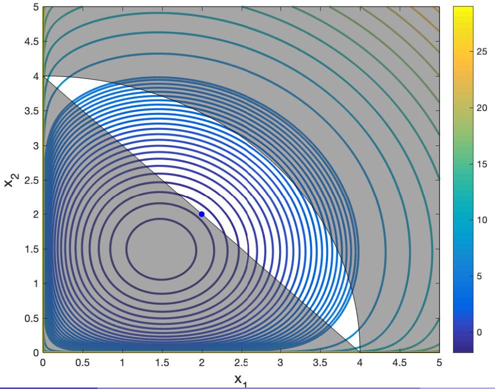
x >= 2



Example: Convex problem 2

$$\min_{\boldsymbol{x}\in\mathbb{R}^2_{++}} \ f(x) = (x_1-1)^2 + (x_2-1)^2 - \sqrt{x_1+x_2} - \ln(x_1) - \ln(x_2)$$

$$\begin{aligned} \text{s.t. } c_1(\boldsymbol{x}) &= x_1 + x_2 - 4 \geq 0 \\ c_2(\boldsymbol{x}) &= -x_1^2 - x_2^2 + 16 \geq 0 \end{aligned}$$



= no constraints.
I can use unstrained solver
(fx steepest descent)

Illustration of optimality conditions

- **Case 0:** No equality constraints, no active inequality constraints.

  We can consider it as unconstrained problem and move in a descent direction. If the step length is not too large, then the constraints are of no consequence.
  Interior local minimizer should satisfy $\nabla f(\boldsymbol{x}^*) = 0$.

- **Case 1:** One equality constraint (no inequality constraints).
  - $\boldsymbol{x}^*$ may be a local, constrained minimizer and it satisfies

$$\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*) \text{ with } \lambda \in \mathbb{R}.$$

- **Case 2:** A single inequality constraint (no equality constraints).

  If no decrease of f is possible:
  - Interior point $c_1(\boldsymbol{x}^*) > 0$ (inactive constraint): $\nabla f(\boldsymbol{x}^*) = 0$.
  - Boundary point $c_1(\boldsymbol{x}^*) = 0$:
    $\boldsymbol{x}^*$ should satisfy $\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*)$ with $\lambda \geq 0$.

Illustration of optimality conditions

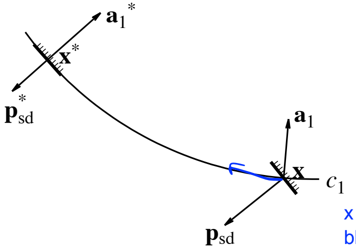
- **Case 0:** No equality constraints, no active inequality constraints.

  Interior local minimizer should satisfy $\nabla f(\boldsymbol{x}^*) = 0$.

- **Case 1:** One equality constraint (no inequality constraints).

min f(x)
s.t. c1(x) = 0

x* is a minimizer:
you cannot move along
line in descent direction



Feasible set:
all points on c1 curve

x Cannot be minimizer:
blue line is descent direction

  - The feasible set is $\Omega = \{\boldsymbol{x} : c_1(\boldsymbol{x}) = 0\}$.
  - $\boldsymbol{p}_{\rm sd} = -\nabla f(\boldsymbol{x})$ is the steepest descent direction.
  - $\boldsymbol{a}_1 = \nabla c_1(\boldsymbol{x})$ is the constraint gradient.
  - $\boldsymbol{x}^*$ may be a local, constrained minimizer and it satisfies

$$\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*) \text{ with } \lambda \in \mathbb{R}.$$

Illustration of optimality conditions

- **Case 0:** No equality constraints, no active inequality constraints.

  Interior local minimizer should satisfy $\nabla f(\boldsymbol{x}^*) = 0$.

- **Case 1:** One equality constraint (no inequality constraints).
  - $\boldsymbol{x}^*$ may be a local, constrained minimizer and it satisfies

$$\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*) \text{ with } \lambda \in \mathbb{R}.$$

- **Case 2:** A single inequality constraint (no equality constraints).

  If no decrease of f is possible:

  min f(x)
  s.t. c(x) >= 0

  - Interior point $c_1(\boldsymbol{x}^*) > 0$ (inactive constraint): $\nabla f(\boldsymbol{x}^*) = 0$.
  - Boundary point $c_1(\boldsymbol{x}^*) = 0$: There cannot be a direction $\boldsymbol{p}$ such that $\nabla f(\boldsymbol{x}^*)^T \boldsymbol{p} < 0$ and $\nabla c_1(\boldsymbol{x}^*)^T \boldsymbol{p} \geq 0$.
    $\boldsymbol{x}^*$ should satisfy $\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*)$ with $\lambda \geq 0$.

Illustration of optimality conditions

- **Case 3:** Two inequality constraints (no equality constraints).

  If no decrease of f is possible:
  - Both are inactive: $\nabla f(\boldsymbol{x}^*) = 0$.
  - One active ($c_1(\boldsymbol{x}^*) = 0$) and the other inactive ($c_2(\boldsymbol{x}^*) > 0$): $\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*)$ with $\lambda \geq 0$.
  - Both active ($c_1(\boldsymbol{x}^*) = c_2(\boldsymbol{x}^*) = 0$):

min f(x)
s.t.
c1(x) >= 0
c2(x) >= 0

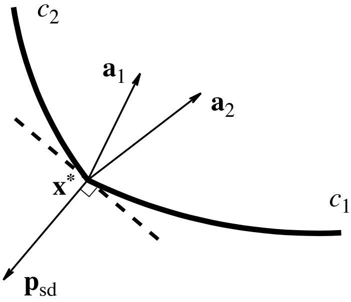


Illustration of optimality conditions

- **Case 3:** Two inequality constraints (no equality constraints).

  If no decrease of f is possible:

  - Both are inactive: $\nabla f(\boldsymbol{x}^*) = 0$.
  - One active ($c_1(\boldsymbol{x}^*) = 0$) and the other inactive ($c_2(\boldsymbol{x}^*) > 0$): $\nabla f(\boldsymbol{x}^*) = \lambda \nabla c_1(\boldsymbol{x}^*)$ with $\lambda \geq 0$.
  - Both active ($c_1(\boldsymbol{x}^*) = c_2(\boldsymbol{x}^*) = 0$): There cannot be a direction $\boldsymbol{p}$ such that $\nabla f(\boldsymbol{x}^*)^T \boldsymbol{p} < 0$ and $\nabla c_i(\boldsymbol{x}^*)^T \boldsymbol{p} \geq 0$ with $i \in \mathcal{I}$.

    $\boldsymbol{x}^*$ should satisfy

$$\nabla f(\boldsymbol{x}^*) = \lambda_1 \nabla c_1(\boldsymbol{x}^*) + \lambda_2 \nabla c_2(\boldsymbol{x}^*) \quad \text{with} \quad \lambda_1, \lambda_2 \geq 0$$

Lagrangian function

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & c_i(\boldsymbol{x}) = 0 \text{ for } i \in \mathcal{E} \\ & c_i(\boldsymbol{x}) \geq 0 \text{ for } i \in \mathcal{I} \end{aligned} \tag{1}$$

Lagrangian function

The Lagrangian function for the problem (1) is defined by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i\in\mathcal{E}\cup\mathcal{I}} \lambda_i c_i(\boldsymbol{x}),$$

where $\{\lambda_i\}$ are the *Lagrangian multipliers*.

- The gradient of $\mathcal{L}$ with respect to $\boldsymbol{x}$ is denoted as

$$\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \nabla f(\boldsymbol{x}) - \sum_{i\in\mathcal{E}\cup\mathcal{I}} \lambda_i \nabla c_i(\boldsymbol{x}).$$

First-order necessary conditions (KKT conditions)

Theorem

Suppose that

1. $\boldsymbol{x}^*$ is a local minimizer of (1), where f and c_i are continuously differentiable;
2. either all active constraints c_i are linear,
   or the gradients $\{\nabla c_i(\boldsymbol{x}^*)\}$ for all active constraints are linearly independent *(LICQ)*. linear independent constrained

Then there exists a Lagrangian multiplier vector $\boldsymbol{\lambda}^*$ such that

1. $\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0$, *(stationary)*
2. $c_i(\boldsymbol{x}^*) = 0$ for $i \in \mathcal{E}$, *(feasibility)*
3. $c_i(\boldsymbol{x}^*) \geq 0$ for $i \in \mathcal{I}$, *(feasibility)*
4. $\lambda_i^* \geq 0$ for $i \in \mathcal{I}$,
5. $\lambda_i^* c_i(\boldsymbol{x}^*) = 0$, for $i \in \mathcal{E} \cup \mathcal{I}$. *(complementarity)*

KKT conditions

- For an equality constraint $c_i(\boldsymbol{x}^*) = 0$, λ_i^* can have any sign.
- For an active inequality constraint $c_i(\boldsymbol{x}^*) \geq 0$, λ_i^* is nonnegative.
- For an inactive inequality constraint $c_i(\boldsymbol{x}^*) > 0$, we must have $\lambda_i^* = 0$.
- If $\mathcal{I} = \emptyset$, then KKT is reduced to $\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0$ together with equality constraints.
- $f(\boldsymbol{x}^*) = \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$.
- **Strict complementarity:** Exactly one of λ_i^* and $c_i(\boldsymbol{x}^*)$ is zero for each $i \in \mathcal{I}$.
- **Constrained stationary point:** $\boldsymbol{x}^*$ is feasible and $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ satisfies the KKT conditions.

Example: Equality constrained problem

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}), \quad f \text{ convex} \\ \text{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

KKT conditions for this problem are

$$\begin{aligned} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*) + A^T \boldsymbol{\lambda}^* &= 0 \\ A\boldsymbol{x}^* - \boldsymbol{b} &= 0 \end{aligned}$$

We can use Newton's method to solve this system of nonlinear equations:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^{\boldsymbol{x}}, \quad \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k \boldsymbol{p}_k^{\boldsymbol{\lambda}},$$

where $\boldsymbol{p}_k^{\boldsymbol{x}}$ and $\boldsymbol{p}_k^{\boldsymbol{\lambda}}$ is obtained by solving

$$\begin{bmatrix} \nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x}_k) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_k^{\boldsymbol{x}} \\ \boldsymbol{p}_k^{\boldsymbol{\lambda}} \end{bmatrix} = - \begin{bmatrix} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k) + A^T \boldsymbol{\lambda}_k \\ A\boldsymbol{x}_k - \boldsymbol{b} \end{bmatrix}$$

Quadratic programs

The quadratic programming problem

$$\begin{aligned}\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{x} = b_i, \quad i\in\mathcal{E} \\ & \boldsymbol{a}_i^T\boldsymbol{x} \geq b_i, \quad i\in\mathcal{I}\end{aligned}$$

- If H is positive semidefinite, it is a convex QP.
- If H is positive definite, it is a strictly convex QP.
- $\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} - \boldsymbol{\lambda}^T(A^T\boldsymbol{x} - \boldsymbol{b})$.

Quadratic programs

The quadratic programming problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x}$$
$$\text{s.t.} \quad \boldsymbol{a}_i^T\boldsymbol{x} = b_i, \quad i \in \mathcal{E}$$

- $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} - \boldsymbol{\lambda}^T(A^T\boldsymbol{x} - \boldsymbol{b})$.

If only equality constraint, KKT conditions are:

$$H\boldsymbol{x}^* + \boldsymbol{g} - A\boldsymbol{\lambda}^* = 0$$
$$A^T\boldsymbol{x}^* - \boldsymbol{b} = 0$$

linear system:
conjugate gradient method

Quadratic programs

The quadratic programming problem

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{x} = b_i, \quad i\in\mathcal{E} \\ & \boldsymbol{a}_i^T\boldsymbol{x} \geq b_i, \quad i\in\mathcal{I} \end{aligned}$$

- $\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} - \boldsymbol{\lambda}^T(A^T\boldsymbol{x} - \boldsymbol{b})$.

In general case, KKT conditions are:

$$\begin{aligned} H\boldsymbol{x}^* + \boldsymbol{g} - \sum_{i\in\mathcal{A}(\boldsymbol{x}^*)} \lambda_i^*\boldsymbol{a}_i &= 0 \\ \boldsymbol{a}_i^T\boldsymbol{x}^* &= b_i \quad i\in\mathcal{A}(\boldsymbol{x}^*) \\ \boldsymbol{a}_i^T\boldsymbol{x}^* &> b_i \quad i\in\mathcal{I}\backslash\mathcal{A}(\boldsymbol{x}^*) \\ \lambda_i^* &\geq 0, \quad i\in\mathcal{I}\cap\mathcal{A}(\boldsymbol{x}^*) \end{aligned}$$

where $\mathcal{A}(\boldsymbol{x}^*) = \{i\in\mathcal{E}\cup\mathcal{I} : \boldsymbol{a}_i^T\boldsymbol{x}^* = b_i\}$ denotes the active set at $\boldsymbol{x}^*$.

Numerical methods for QP

Mainly three types:

- **Active-set methods.**
  - **Idea:** Identify the optimal active set from an initial guess for it, by repeatedly adding or subtracting one constraint each time.
  - Appropriate for small- or medium-scale problems; particularly for convex QP.
- **Gradient projection methods.**
  - **Idea:** Apply the steepest descent method but "bending" along the constraints.
  - Appropriate for large-scale problems; particularly simple with box constraints.
- **Interior-point methods.**
  - **Idea:** Apply Newton-like step on the KKT system.
  - Appropriate for large-scale problems.

`cvxopt.solvers.qp`

- CVXOPT is a free software package for convex optimization based on the Python programming language.
- `cvxopt.solvers` includes convex optimization routines and optional interfaces to solvers from GLPK, MOSEK, and DSDP5.
- `cvxopt.solvers.qp` attempts to solve the quadratic programming problem:

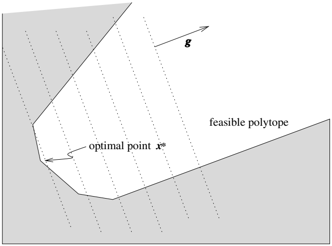
$$\min_{\boldsymbol{x}} \ \frac{1}{2}\boldsymbol{x}^T P \boldsymbol{x} + \mathrm{q}^T \boldsymbol{x} \qquad \text{subject to: } G\boldsymbol{x} \le \mathrm{h} \text{ and } A\boldsymbol{x} = \boldsymbol{b}$$

- `cvxopt.solvers.qp(P, q[, G, h[, A, b[, solver[, initvals]]]])`
  - `solvers.options` can be used to set the maximum number of iterations, tolerances, etc.

Linear programs

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}) = \boldsymbol{g}^T\boldsymbol{x}$$
$$\text{s.t.} \quad A\boldsymbol{x} = \boldsymbol{b}$$
$$C\boldsymbol{x} \geq \boldsymbol{d}$$

- The objective function is linear.
- The constraints are linear.
- It is convex as well as concave.
- The feasible set is a polytope.
- The number of minimizers: 0 (infeasible or unbounded), 1 (a vertex) or ∞ (a face).
- $\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{s}) = \boldsymbol{g}^T\boldsymbol{x} - \boldsymbol{\lambda}^T(A\boldsymbol{x}-\boldsymbol{b}) - \boldsymbol{s}^T(C\boldsymbol{x}-\boldsymbol{d})$.



Linear programs

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) = \boldsymbol{g}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & C\boldsymbol{x} \geq \boldsymbol{d} \end{aligned}$$

- $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{s}) = \boldsymbol{g}^T\boldsymbol{x} - \boldsymbol{\lambda}^T(A\boldsymbol{x} - \boldsymbol{b}) - \boldsymbol{s}^T(C\boldsymbol{x} - \boldsymbol{d})$.

KKT conditions are:

$$\begin{aligned} A^T\boldsymbol{\lambda}^* + C^T\boldsymbol{s}^* &= \boldsymbol{g} \\ A\boldsymbol{x}^* &= \boldsymbol{b} \\ C\boldsymbol{x}^* &\geq \boldsymbol{d} \\ \boldsymbol{s}^* &\geq 0 \\ {\boldsymbol{s}^*}^T(C\boldsymbol{x}^* - \boldsymbol{d}) &= 0 \end{aligned}$$

Linear programs

$$
\begin{aligned}
\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) = \boldsymbol{g}^T\boldsymbol{x} \\
\text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\
& C\boldsymbol{x} \geq \boldsymbol{d}
\end{aligned}
$$

Main type of methods:

- **Simplex method:** an active-set method.
- **Interior-point methods:** Apply Newton-like step on the KKT system. Appropriate for large-scale problems.

scipy.optimize.linprog OR cvxopt.solvers.lp

- Both attempt to solve the linear programming problem:

$$\min_{\boldsymbol{x}} \; \mathrm{c}^T \boldsymbol{x} \qquad \text{subject to: } G\boldsymbol{x} \leq \mathrm{h} \text{ and } A\boldsymbol{x} = \boldsymbol{b}$$

- `scipy.optimize.linprog(c, A_ub=G, b_ub=h, A_eq=A, b_eq=b, bounds=(0, None), method='highs', callback=None, options=None, x0=None, integrality=None)`
  - `bounds` defines the range of $\boldsymbol{x}$;
  - `method` by default applies the high-performance parallel linear programming software, HiGHS;
  - `integrality` indicates the type of integrality constraints on $\boldsymbol{x}$.

- `cvxopt.solvers.lp(c, G, h[, A, b[, solver[, primalstart[, dualstart]]]])`
  - `solvers.options` can be used to set the maximum number of iterations, tolerances, etc.

scipy.optimize.minimize

scipy.optimize.

minimize

```
minimize(fun, x0, args=(), method=None, jac=None, hess=None, hessp=None,
bounds=None, constraints=(), tol=None, callback=None, options=None)
```

Minimization of scalar function of one or more variables. [source]

Inputs:

- fun: the objective function
- x0: an starting point;
- method: to specify the choice of the optimization methods, e.g. 'SLSQP', 'trust-constr', etc;
- bounds: to specify the the bound of $\boldsymbol{x}$;
- constraints: constraints definition.
  - Include linear constraints and nonlinear constraints.
  - Depending on the method, constraints need be defined as a single object or a list of dictionaries.

Optimization related courses

- 02611 Optimization for Data Science
  - Master course. Every spring.
- 02612 Constrained Optimization
  - Master course. Every spring.
- 02947 PDE constrained optimization
  - PhD course. Every odd year.
- Optimization technique is also applied in many other courses, such as model predictive control (02619), inverse problems and imaging (02624), machine learning for signal processing (02471), mathematical modelling (02526), image analysis (02502), etc..

Final evaluation
in DTUinside
from 18. Nov. to 29. Nov.