Lecture 3 : Optimization Terminology

goals:

- introduce basic terminology for optimization problems

Mathematical Optimization Problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in \mathcal{X} \end{aligned}$$

- $x \in \mathbb{R}^n$: (decision) optimization variables
- $f : \mathbb{R}^n \to \mathbb{R}$: objective/cost function
- $\mathcal{X} \subseteq \mathbb{R}^n$: feasible/constraint set

$$\mathcal{X} = \{x \in \mathbb{R}^n \mid g_i(x) \le 0,\ i=1,\dots,m,\ h_i(x) = 0,\ i=1,\dots,p\}$$

$g_i(x)$: inequality constraint functions

$h_i(x)$: equality constraint functions

- aka a nonlinear program

• Note: all other parameters used to specify objective and constraint functions are referred to as <u>problem data</u>

<u>Ex</u>

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Ax \preceq b \end{aligned}$$

Variable: $x \in \mathbb{R}^n$

problem data: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$

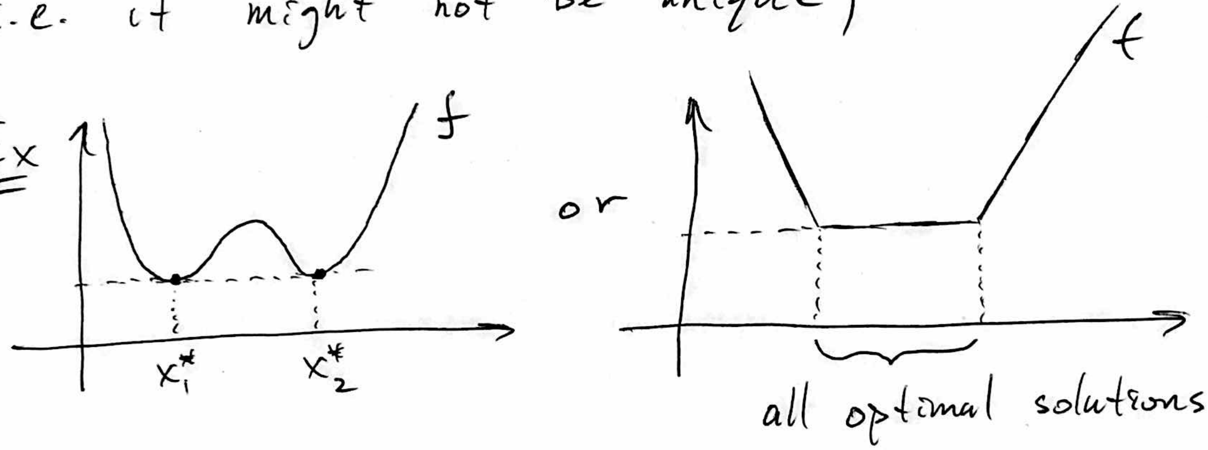
- Note: any <u>maximization</u> problem can be written this way by changing the sign of the objective

- an <u>optimal solution</u> x^* has the smallest value of f among all vectors satisfying the constraints

$$x^* \in \mathcal{X} \quad \text{satisfies} \quad f(x^*) \le f(x) \quad \forall x \in \mathcal{X}$$

- there may be more than one optimal solution (i.e. it might not be unique)

Ex



the set of all optimal solutions is written

$$\underset{x \in \mathcal{X}}{\operatorname{argmin}}\ f(x) = \left\{ x \in \mathcal{X} \;\middle|\; f(x) = \min_{x \in \mathcal{X}} f(x) \right\}$$

and we can write $x^* \in \underset{x \in \mathcal{X}}{\operatorname{argmin}}\ f(x)$

- the <u>optimal value</u> is $f^* = \min_{x \in \mathcal{X}} f(x)$

- It's also possible an optimal solution <u>does not exist</u>

① If X is empty ($X = \phi$)
the problem is called <u>infeasible</u>

- In this case, we set $f^* = +\infty$

- This means the constraints are inconsistent, i.e. there's no point that satisfies all of them

<u>Ex</u>

$$\min_{x \in \mathbb{R}} \quad x^2$$

$$\text{subject to} \quad x \leq -1$$
$$x \geq 1$$

- In control design, this means the design specifications are too strict

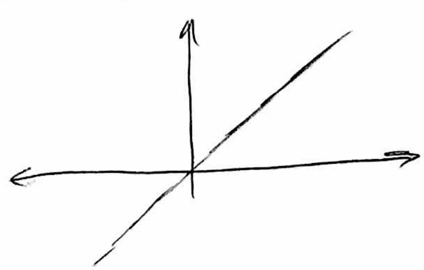
② If $f(x)$ can be made arbitrarily negative without violating any constraints, the problem is <u>unbounded</u> (below)

- In this case, we set $f^* = -\infty$

<u>Ex</u>

$$\min_{x \in \mathbb{R}} \quad x$$

$$\text{subject to} \quad x \leq 0$$



- Usually means the problem is <u>ill-posed</u> as an engineering design problem

③ $f(x)$ might be bounded below, but there exists no point that achieves the bound

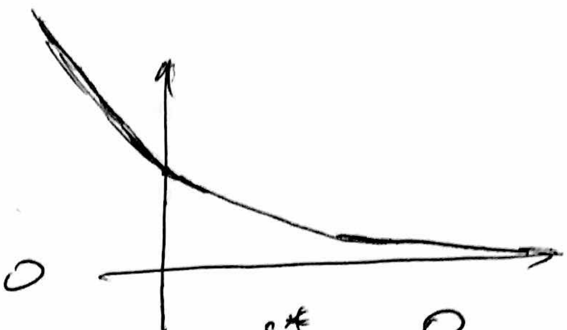
- Here we set $f^* = \underset{x \in X}{\text{infimum}} \; f(x)$

greatest lower bound

- The optimal <u>value</u> exists but an optimal <u>solution</u> does not

<u>Ex</u> $\inf_{x \in \mathbb{R}} \; e^{-x}$

subject to $x \geq 0$



$f^* = 0$

- Two cases where x^* guaranteed to exist

① If f is continuous and X is nonempty and <u>compact</u> (i.e, closed and bounded)

- Weierstrass Extreme Value Theorem (1830)

② If f is continuous and <u>radially</u> <u>unbounded</u>

i.e. $\lim_{\|x\| \to \infty} f(x) = \infty$

- Feasible point : A vector $x \in X$ satisfying the inequality and equality constraints

- Strictly feasible point : A vector $x \in X$ satisfying the inequality constraints strictly : $g_i(x) < 0 \quad i=1,\dots,m$

- If $X = \mathbb{R}^n$ (i.e. there are no constraints) then the problem is called unconstrained

Let $\bar{x} \in X$ and $B(\bar{x}, \varepsilon) = \{ x \mid \|x - \bar{x}\| \leq \varepsilon \}$

The point $\bar{x}$ is called a

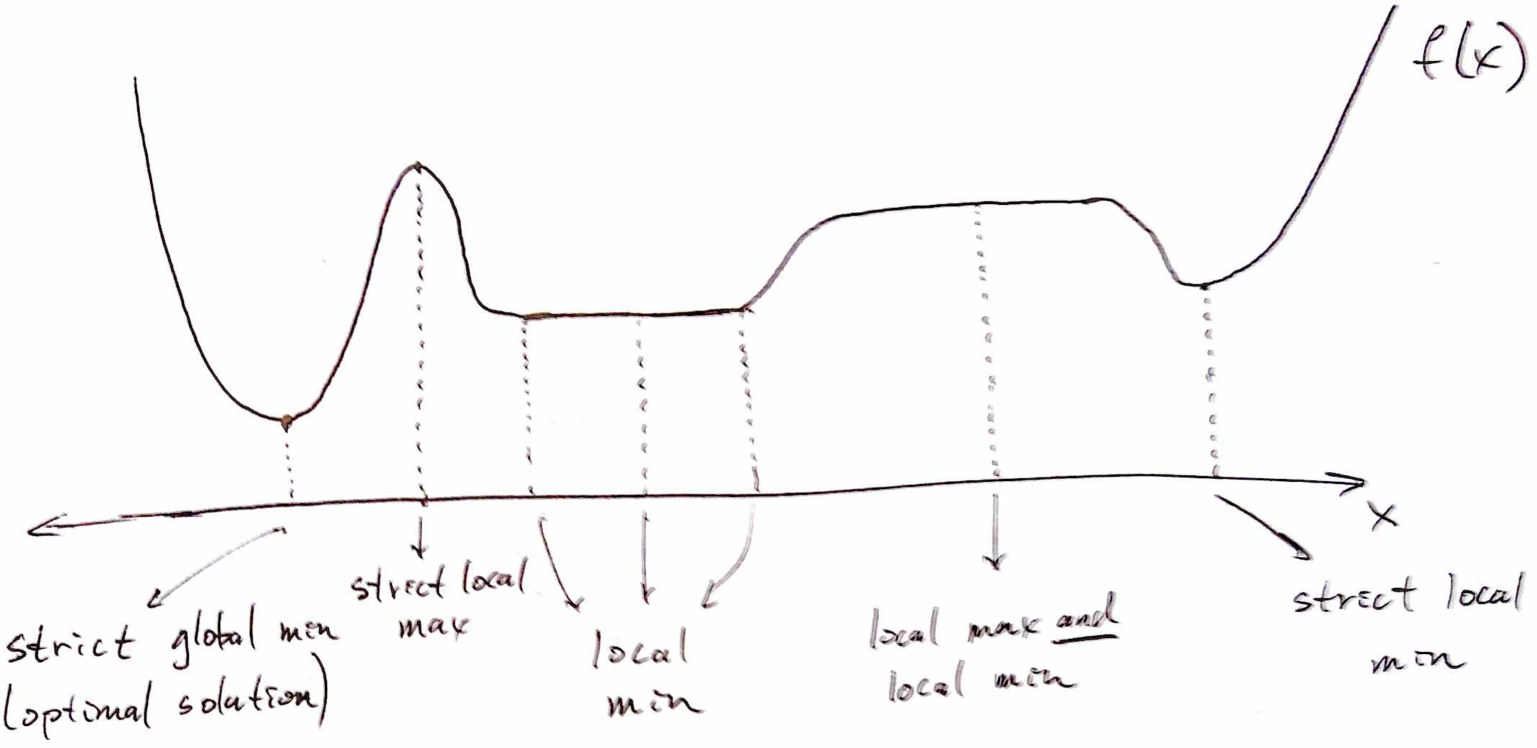
- local minimum if $\bar{x} \in X$ and $\exists \varepsilon > 0$ such that
$$f(\bar{x}) \leq f(x) \quad \forall x \in B(\bar{x}, \varepsilon) \cap X$$

- strict local minimum if $\bar{x} \in X$ and $\exists \varepsilon > 0$ s.t.
$$f(\bar{x}) < f(x) \quad \forall x \in B(\bar{x}, \varepsilon) \cap X, \; x \neq \bar{x}$$

- global minimum if $\bar{x} \in X$ and $f(\bar{x}) \leq f(x) \quad \forall x \in X$

- strict global minimum if $\bar{x} \in X$ and $f(\bar{x}) \leq f(x) \quad \forall x \in X \quad x \neq \bar{x}$

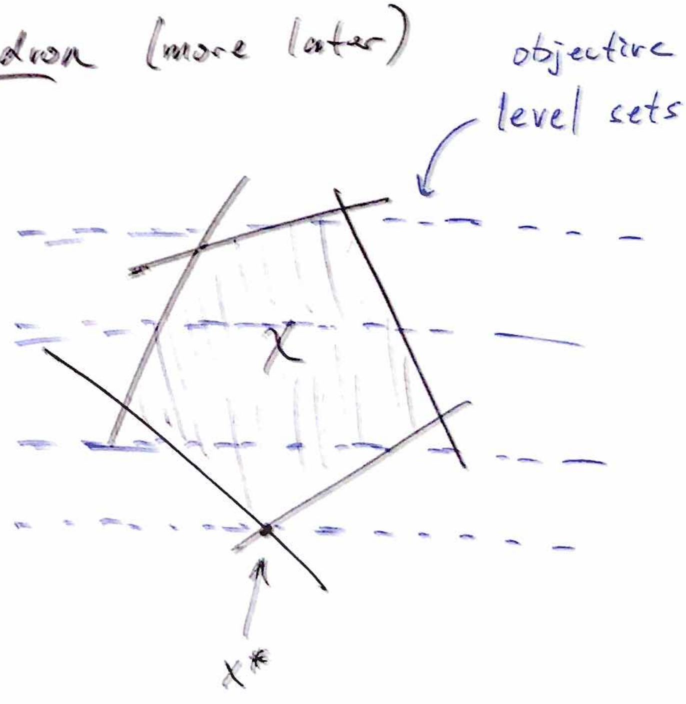
- saddle point if it's a strict local minimum restricted along at least one direction and a strict local maximum along at least one other direction



Some Common Optimization Problems

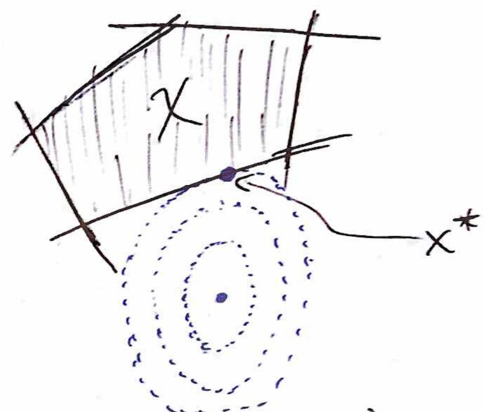
- Linear Program (LP)
  - linear cost and constraint functions
  - feasible set is a polyhedron (more later)

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Gx \le h \\ & Ax = b \end{aligned}$$



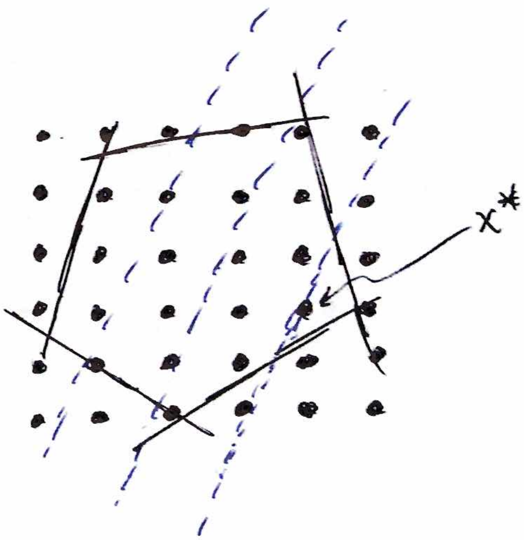
- Quadratic Program (QP)
    - quadratic cost, linear constraints
    - feasible set is a polyhedron
    - convex if $P \succeq 0$, nonconvex otherwise (more later)

$$\begin{aligned} \text{minimize} \quad & x^T P x + q^T x \\ \text{subject to} \quad & Gx \leq h \\ & Ax = b \end{aligned}$$



- Mixed Integer Linear Program (MILP)
    - LP with binary or integer constraints
    - nonconvex (more later)

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Gx \leq h \\ & Ax = b \\ & x \in \{0,1\}^n \text{ or } x \in \mathbb{Z}^n \end{aligned}$$



- Others: second-order cone program, semidefinite program, sum of squares program, geometric program, robust or stochastic program, etc.

Optimization Algorithms & Software

- <u>Huge</u> range of algorithms for many types of problems (both convex + nonconvex)
    - Check out Nocedal + Wright book to get an idea of breadth
        - gradient descent + many variations, Newton methods + many variations, line search + trust region methods, simplex, active set, interior point, augmented Lagrangian, projected + proximal, etc. etc.
    - A highly active research area

- <u>Huge</u> range of software implementations, both proprietary + open source
    - Matlab (linprog/quadprog), CPLEX (IBM), Gurobi, GLPK, XPRESS, OOQP, FORCES, SDPT3, Sedumi, Mosek, IPOPT, ECOS, etc. etc.
    - No standard solver interface
    - Modelling tools/parsers allow easy switching b/t solvers
        - GAMS, AMPL, CVX, YALMIP, CVXpy, CVX.jl, etc
        - slower than direct solvers, but great for rapid prototyping in research