4. Optimization problems

- terminology
- problem transformations
- solving optimization problems
- control example

Optimization problem

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & g_i(\boldsymbol{x}) \le 0, \quad i = 1, \ldots, m \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, p \end{array} \tag{4.1}$$

- $\boldsymbol{x} = (x_1, \ldots, x_n)$ is the *optimization variable* or *decision variable*
- $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function* or *cost function*
- $g_i(\boldsymbol{x}) \le 0$ are the *inequality constraints* ($g_i : \mathbb{R}^n \to \mathbb{R}$)
- $h_j(\boldsymbol{x}) = 0$ are the *equality constraints* ($h_j : \mathbb{R}^n \to \mathbb{R}$)
- the problem is said to be *unconstrained* if $g_i = h_j = 0$

Solution: a point $\boldsymbol{x}^\star$ is an *optimal point* or *solution* to (4.1) if

$$f(\boldsymbol{x}^\star) \le f(\boldsymbol{x})$$

for any $\boldsymbol{x}$ with $g_1(\boldsymbol{x}) \le 0, \ldots, g_m(\boldsymbol{x}) \le 0$ and $h_1(\boldsymbol{x}) = 0, \ldots, h_p(\boldsymbol{x}) = 0$

Compact representation

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & g(\boldsymbol{x}) \leq \mathbf{0} \\ & h(\boldsymbol{x}) = \mathbf{0} \end{array} \tag{4.2}$$

- $g(\boldsymbol{x}) = (g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x}))$
- $h(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \ldots, h_p(\boldsymbol{x}))$

Maximization problem

$$\begin{array}{ll} \text{maximize} & f(\boldsymbol{x}) \\ \text{subject to} & g(\boldsymbol{x}) \leq \mathbf{0},\ h(\boldsymbol{x}) = \mathbf{0} \end{array} \tag{4.3}$$

- a vector $\boldsymbol{x}^\star$ is *optimal* or a *solution* if it maximizes the objective:

  $$f(\boldsymbol{x}^\star) \geq f(\boldsymbol{x})$$

  for all $\boldsymbol{x}$ satisfying $g(\boldsymbol{x}) \leq \mathbf{0}$ and $h(\boldsymbol{x}) = \mathbf{0}$
- f is often called *utility function* instead of cost

Maximization as minimization: we have

$$\max\ f(\boldsymbol{x}) = -\min\ -f(\boldsymbol{x})$$

hence

$$\begin{array}{ll} \text{maximize} & f(\boldsymbol{x}) \\ \text{subject to} & g(\boldsymbol{x}) \leq \mathbf{0},\ h(\boldsymbol{x}) = \mathbf{0} \end{array} \quad \leftrightarrow \quad \begin{array}{ll} \text{minimize} & -f(\boldsymbol{x}) \\ \text{subject to} & g(\boldsymbol{x}) \leq \mathbf{0},\ h(\boldsymbol{x}) = \mathbf{0} \end{array}$$

- both problems share the same solutions
- the optimal value of one is the negative of the other
- maximization problems can be treated a minimization problems

Standard form

we refer to problem (4.1) (or (4.2)), namely

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & g_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, m \\ & h_j(\boldsymbol{x}) = 0,\ j = 1, \ldots, p \end{array}$$

as an optimization problem in *standard form* (the righthand side of the inequality and equality constraints are zero)

- we can represent equality constraints $r_j(\boldsymbol{x}) = \tilde{r}_j(\boldsymbol{x})$ as $h_j(\boldsymbol{x}) = 0$ where $h_j(\boldsymbol{x}) = r_j(\boldsymbol{x}) - \tilde{r}_j(\boldsymbol{x})$
- we can express inqualities of the form $\tilde{g}_i(\boldsymbol{x}) \geq 0$ as $g_i(\boldsymbol{x}) \leq 0$ where $g_i(\boldsymbol{x}) = -\tilde{g}_i(\boldsymbol{x})$
- maximization problem can be represented as minimization by changing the objective sign

Example 4.1

the problem

$$\begin{array}{ll} \text{maximize} & -x_1^2 + x_2^2 \\ \text{subject to} & -x_1 + x_2 \geq 10 \\ & x_2 = 2 - x_1 \end{array}$$

can be expressed in standard form with objective

$$\begin{array}{ll} \text{minimize} & x_1^2 - x_2^2 \\ \text{subject to} & x_1 - x_2 + 10 \leq 0 \\ & x_1 + x_2 - 2 = 0 \end{array}$$

Domain and implicit constraints

the *domain* of the optimization problem is the set of points for which the objective and all constraint functions are defined

$$\mathcal{D} = \operatorname{dom} f \ \cap \ \bigcap_{i=1}^{m} \operatorname{dom} g_i \ \cap \ \bigcap_{j=1}^{p} \operatorname{dom} h_j$$

- the standard from problem (4.1) has an *implicit constraint* $\boldsymbol{x} \in \mathcal{D}$
- explicit constraints: $g_1(\boldsymbol{x}) \leq 0, \ldots, g_m(\boldsymbol{x}) \leq 0$, $h_1(\boldsymbol{x}) = 0, \ldots, h_p(\boldsymbol{x}) = 0$
- for example, the problem

  $$\text{minimize} \quad -\log x_1 + \log(x_2 - x_1)$$

  is an unconstrained problem with implicit constraints $x_1 > 0$, $x_2 - x_1 > 0$

Optimal value

Feasible points

- a point $\hat{\boldsymbol{x}}$ is a *feasible point* if $\hat{\boldsymbol{x}} \in \mathcal{D}$ and it satisfies the constraints, $h(\hat{\boldsymbol{x}}) = \mathbf{0}$ and $g(\hat{\boldsymbol{x}}) \leq \mathbf{0}$; otherwise, it is called *infeasible point*
- the problem is said to be *feasible* if there exists at least one feasible point; otherwise, it is said to be *infeasible*

Optimal value: the *optimal value* of the minimization problem (4.2) , denoted by $p^\star$, is the greatest α such that $\alpha \leq f(\boldsymbol{x})$ for all feasible $\boldsymbol{x}$

- for maximization problems, it is the smallest α such that $\alpha \geq f(\boldsymbol{x})$
- if there exists an optimal point $\boldsymbol{x}^\star$, we say the optimal value is *attained* or *achieved* and the problem is *solvable*; in this case, we have $p^\star = f(\boldsymbol{x}^\star)$
- a minimization problem is *unbounded below* if $p^\star = -\infty$; if a minimization problem is infeasible, then we let $p^\star = +\infty$
- a maximization problem is *unbounded above* if $p^\star = \infty$; if a maximization problem is infeasible, then we let $p^\star = -\infty$

Example 4.2

- consider the unconstrained problem

$$\text{minimize} \quad (x_1 - 1)^2 + (x_2 - 1)^2 = \|\boldsymbol{x} - \mathbf{1}\|^2$$

  the optimal value is $p^\star = 0$ attained at the optimal point $\boldsymbol{x}^\star = (1, 1) = \mathbf{1}$

- the problem

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 \\ \text{subject to} & -x_1 \leq 10 \\ & \quad x_2 \geq 0 \end{array}$$

  has solution $\boldsymbol{x}^\star = (-10, 0)$ and $p^\star = -10$

- the problem

$$\text{minimize} \quad x_1^2 - x_2^2$$

  is unbounded below since $f(\boldsymbol{x}) \to -\infty = p^\star$ as $|x_2| \to \infty$

- consider minimizing $f(x) = e^{-x}$; for this problem $p^\star = 0$, but the optimal value is not attained since it only holds as $x \to \infty$

- consider minimizing $f(x) = 1/x$ with domain $\operatorname{dom} f = \{x \mid x > 0\}$; for this problem $p^\star = 0$ but is not attained by any feasible x

- the problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 + x_2 \leq 1 \\ & x_1 + x_2 \geq 2 \end{array}$$

  is an infeasible minimization problem; hence, $p^\star = \infty$

Set-constrained problems

$$\underset{\boldsymbol{x}\in\mathcal{X}}{\text{minimize}} \quad f(\boldsymbol{x})$$

- describes the optimization problem of finding an $\boldsymbol{x} = (x_1, \ldots, x_n)$ that minimizes $f(\boldsymbol{x})$ among all points in the *constraint set* $\mathcal{X} \subset \mathbb{R}^n$
- for problem (4.1), the constraint set is described by *functional constraints*

$$\mathcal{X} = \{\boldsymbol{x} \mid g_i(\boldsymbol{x}) \leq 0,\ h_j(\boldsymbol{x}) = 0,\ i = 1, \ldots, p,\ j = 1, \ldots, m\}$$

- this is not always the case, for example consider the integer set

$$\mathcal{X} = \{1, 2, 3\} \subset \mathbb{R}$$

Simple problems solution

- general optimization problems require sophisticated methods to solve them that utilize derivatives, linear equations, nonlinear operators,...etc
- that said, there are some simple optimization problems that can be solved by inspection or using some basic inequalities such as Cauchy-Schwarz

Example

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{x} - \mathbf{1}\| \\ \text{subject to} & -\mathbf{1} \leq \boldsymbol{x} \leq \mathbf{0} \end{array}$$

- we seek to find a feasible $\boldsymbol{x}$ that is closest in distance to $\mathbf{1}$
- hence

$$\boldsymbol{x}^\star = \mathbf{0} \quad \text{and} \quad p^\star = \|-\mathbf{1}\| = \sqrt{n}$$

Another example:

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 \leq 1 \end{array}$$

- using Cauchy-Schwarz, we can lower bound the objective by

  $$x_1 + x_2 = \mathbf{1}^T \boldsymbol{x} \geq -\|\mathbf{1}\|\|\boldsymbol{x}\| \geq -\sqrt{2}$$

  for all $x_1^2 + x_2^2 \leq 1$
- the minimum value is attained at $\boldsymbol{x} = (-1/\sqrt{2}, -1/\sqrt{2})$, which is feasible
- hence, the optimal point is $\boldsymbol{x} = (-1/\sqrt{2}, -1/\sqrt{2})$

Outline

- terminology
- **problem transformations**
- solving optimization problems
- control example

Equivalent optimization problems

two optimization problems are said to be *equivalent* if from a solution of one, we can find a solution of the other, and vice versa

- for example, maximization problems are equivalent to minimization problems (as shown before)
- many optimization problems can be transformed into equivalent optimization problems that are easier to solve
- it is often useful to look for such equivalence to simplify the task of solving such problems

Scaling optimization problems

$$\begin{array}{ll} \text{minimize} & \alpha f(\boldsymbol{x}) \\ \text{subject to} & \beta_i g_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, m \\ & \gamma_j h_j(\boldsymbol{x}) = 0,\ j = 1, \ldots, p \end{array} \tag{4.4}$$

where $\alpha > 0$, $\beta_i > 0$ and $\gamma_j \neq 0$

- this problem is just a scaled version of (4.1)
- the feasible sets problems (4.4) and (4.1) are identical
- a point $\boldsymbol{x}$ is optimal for the original problem (4.1) if and only if it is optimal for the scaled problem (4.4)
- hence, the two problems are equivalent

Slack variables

note that

$$g_i(\boldsymbol{x}) \leq 0 \text{ if and only if } g_i(\boldsymbol{x}) + s_i = 0$$

for some $s_i \geq 0$

hence, problem (4.1) is equivalent to

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & s_i \geq 0,\ i = 1, \ldots, m \\ & g_i(\boldsymbol{x}) + s_i = 0,\ i = 1, \ldots, m \\ & h_j(\boldsymbol{x}) = 0,\ j = 1, \ldots, p \end{array} \tag{4.5}$$

- the variables are $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{s} \in \mathbb{R}^m$
- the new variable s_i is called the *slack variable* associated with the inequality constraint $g_i(\boldsymbol{x}) \leq 0$

Monotone transformations

$$\begin{array}{ll} \text{minimize} & \phi(f(\boldsymbol{x})) \\ \text{subject to} & g_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, m \\ & h_j(\boldsymbol{x}) = 0,\ j = 1, \ldots, p \end{array} \tag{4.6}$$

- $\phi : \mathbb{R} \to \mathbb{R}$ is a continuous and monotone increasing function (*i.e.*, $\phi(a) > \phi(b)$ for all $a > b$) over the feasible set $\mathcal{X}$
- note that ϕ is one-to-one and its inverse ϕ^{-1} is well defined
- problem (4.6) is equivalent to (4.1)

Constraint transformation: functions $\psi_i : \mathbb{R} \to \mathbb{R}$ and $\varphi_j : \mathbb{R} \to \mathbb{R}$ can be used to transform the constraints into equivalent ones if

- $\psi_i(g_i(\boldsymbol{x})) \leq 0$ if and only if $g_i(\boldsymbol{x}) \leq 0$
- $\varphi_j(h_j(\boldsymbol{x})) = 0$ if and only if $h_j(\boldsymbol{x}) = 0$

Example 4.3

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{x}\| \\ \text{subject to} & g(\boldsymbol{x}) \leq \mathbf{0} \end{array}$$

the norm is always nonnegative and the square function $\phi(\cdot) = (\cdot)^2$ is monotone increasing over nonnegative numbers, hence, we can transform the problem into

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{x}\|^2 \\ \text{subject to} & g(\boldsymbol{x}) \leq \mathbf{0} \end{array}$$

- the two problems are equivalent since the optimal points are the same
- unlike the original problem the new objective function is differentiable, which simplifies the problem

Change of variables

suppose that $F : \mathcal{X} \to \mathcal{Y}$ is a one-to-one function, *i.e.*, for every $\boldsymbol{y} \in \mathcal{Y}$, there exists a unique $\boldsymbol{x} \in \mathcal{X}$ such that

$$\boldsymbol{y} = F(\boldsymbol{x}) \iff \boldsymbol{x} = F^{-1}(\boldsymbol{y})$$

in this case, problem (4.1) is equivalent to

$$\begin{array}{ll} \text{minimize} & f\big(F^{-1}(\boldsymbol{y})\big) \\ \text{subject to} & g_i\big(F^{-1}(\boldsymbol{y})\big) \leq 0,\ i = 1, \ldots, m \\ & h_j\big(F^{-1}(\boldsymbol{y})\big) = 0,\ j = 1, \ldots, p \end{array}$$

Example 4.4

$$\begin{array}{ll} \text{minimize} & x_1 x_2 x_3^2 \\ \text{subject to} & x_1 x_2 \leq 2 \\ & x_1, x_2, x_3 > 0 \end{array}$$

since $\log(\cdot)$ is strictly increasing (for non-negative argument), we can use monotone transformations $\log(x_1 x_2 x_3^2)$ and $\log(x_1 x_2) \leq \log(2)$ and the change of variable $z_i = \log x_i$ to transform the problem into

$$\begin{array}{ll} \text{minimize} & z_1 + z_2 + 2z_3 \\ \text{subject to} & z_1 + z_2 \leq \log 2 \end{array}$$

the above problem is linear, which is easier to solve than the original nonlinear formulation

Example 4.5

$$\begin{array}{ll} \text{minimize} & x_1x_2 - x_3^2 \\ \text{subject to} & x_1 + x_2 + x_3 \leq 20 \\ & x_2 \geq 10 \end{array}$$

if we let $y_1 = (x_1 + x_2)/2$ and $y_2 = (x_1 - x_2)/2$ then

$$x_1 = y_1 + y_2, \quad x_2 = y_1 - y_2$$

using the above change of variables, we can transform the problem into

$$\begin{array}{ll} \text{minimize} & y_1^2 - y_2^2 - x_3^2 \\ \text{subject to} & 2y_1 + x_3 \leq 20 \\ & y_1 - y_2 \geq 10 \end{array}$$

notice that the objective is now separable in the new variables; for separable problems, there exist efficient specialized optimization methods

Eliminating equality constraints

let the function $\phi : \mathbb{R}^k \to \mathbb{R}^n$ be defined such that $\boldsymbol{x}$ satisfies $h(\boldsymbol{x}) = \mathbf{0}$ if and only if there is some $\boldsymbol{z}$ such that $\boldsymbol{x} = \phi(\boldsymbol{z})$, then problem (4.1) is equivalent to

$$\begin{array}{ll} \text{minimize} & f(\phi(\boldsymbol{z})) \\ \text{subject to} & g_i(\phi(\boldsymbol{z})) \leq 0, \ i = 1, \ldots, m \end{array} \tag{4.7}$$

- if $\boldsymbol{z}$ is optimal for the transformed problem, then $\boldsymbol{x} = \phi(\boldsymbol{z})$ is optimal for the original problem
- conversely, if $\boldsymbol{x}$ is optimal for the original problem, then (since $\boldsymbol{x}$ is feasible) then any $\boldsymbol{z}$ such that $\boldsymbol{x} = \phi(\boldsymbol{z})$ is optimal for the transformed problem

Example 4.6

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 + x_3^2 \\ \text{subject to} & x_1 - x_2 x_3 = 1 \end{array}$$

we can use $x_1 = 1 + x_2 x_3$ to remove the equality constraint and get the equivalent problem

$$\text{minimize} \quad 1 + x_2 x_3 + x_2 + x_3^2$$

- in this case, we have $\phi(z_1, z_2) = (1 + z_1 z_2, z_1, z_2)$
- this problem is unconstrained, which is easier to solve than the original problem

Example 4.7

$$\begin{array}{ll} \text{minimize} & x_1 + 4x_2 + x_3 \\ \text{subject to} & 2x_1 - 2x_2 + x_3 = 4 \\ & x_1 - x_3 = 1 \\ & x_2 \geq 0,\ x_3 \geq 0 \end{array}$$

using the equality constraints, we have $x_1 = 1 + x_3$ and

$$2x_1 - 2x_2 + x_3 = 2(1 + x_3) - 2x_2 + x_3 = 4 \Rightarrow x_2 = \frac{3}{2}x_3 - 1$$

hence, the problem can be simplified to

$$\begin{array}{ll} \text{minimize} & 8x_3 + 3 \\ \text{subject to} & x_3 \geq 2/3 \end{array}$$

with solution $x_3 = 2/3$; putting things together the solution to the original problem is $\boldsymbol{x}^\star = (5/3, 0, 2/3)$

Removing linear constraints

$$h(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$$

where $A \in \mathbb{R}^{p\times n}$ and $\boldsymbol{b} \in \mathbb{R}^p$; assume that the first p columns of A are linearly independent so that

$$A = [B\ D]$$

- B is an invertible $p \times p$ matrix and D is an $p \times (n-p)$ matrix
- recall that the solutions of $A\boldsymbol{x} = \boldsymbol{b}$ can parametrized by

  $$\boldsymbol{x} = \hat{\boldsymbol{x}} + F\boldsymbol{z}$$

  for any arbitrary $\boldsymbol{z} \in \mathbb{R}^{(n-p)}$ where

  $$\hat{\boldsymbol{x}} = \begin{bmatrix} B^{-1}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}, \quad F = \begin{bmatrix} -B^{-1}D \\ I \end{bmatrix}$$

  the columns of the matrix F form a basis for the nullspace of A $(\mathrm{range}(F) = \mathrm{null}(A))$

substituting $\boldsymbol{x} = \hat{\boldsymbol{x}} + F\boldsymbol{z}$ into the original problem gives

$$\begin{array}{ll} \text{minimize} & f(\hat{\boldsymbol{x}} + F\boldsymbol{z}) \\ \text{subject to} & g_i(\hat{\boldsymbol{x}} + F\boldsymbol{z}) \leq 0, \quad i = 1, \ldots, m \end{array}$$

with variable $\boldsymbol{z}$

Adding equality constraints: sometimes it is useful to introduce equality constraints; for example, consider

$$\text{minimize} \quad f(A\boldsymbol{x} + \boldsymbol{b})$$

where $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$; the above problem is equivalent to

$$\text{minimize} \quad f(\boldsymbol{z}) \quad \text{s.t. } \boldsymbol{z} = A\boldsymbol{x} + \boldsymbol{b}$$

with variables $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{z} \in \mathbb{R}^m$

Example 4.8

$$\begin{array}{ll} \text{minimize} & f(x_1, \ldots, x_n) \\ \text{subject to} & x_1 + \cdots + x_n = b \end{array}$$

we can eliminate any x_i, we choose x_n:

$$x_n = b - x_1 - \cdots - x_{n-1}$$

the above corresponds to the choice

$$\hat{\boldsymbol{x}} = b\boldsymbol{e}_n, \quad F = \begin{bmatrix} I \\ -\mathbf{1}^T \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}$$

the transformed problem is

$$\text{minimize} \quad f(x_1, x_2, \ldots, b - x_1 - \cdots - x_{n-1})$$

Optimizing over some variables

it holds that (with abuse of $\inf$ notation):

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}) = \min_{\boldsymbol{x}} \tilde{f}(\boldsymbol{x})$$

where $\tilde{f}(\boldsymbol{x}) = \min_{\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y})$

Example:

$$\begin{array}{ll} \text{minimize} & \boldsymbol{x}_1^T Q_{11} \boldsymbol{x}_1 + 2\boldsymbol{x}_1^T Q_{12} \boldsymbol{x}_2 + \boldsymbol{x}_2^T Q_{22} \boldsymbol{x}_2 \\ \text{subject to} & g_i(\boldsymbol{x}_1) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where Q_{11} and Q_{22} are symmetric; we can analytically minimize over $\boldsymbol{x}_2$:

$$\min_{\boldsymbol{x}_2} \left(\boldsymbol{x}_1^T Q_{11} \boldsymbol{x}_1 + 2\boldsymbol{x}_1^T Q_{12} \boldsymbol{x}_2 + \boldsymbol{x}_2^T Q_{22} \boldsymbol{x}_2\right) = \boldsymbol{x}_1^T \left(Q_{11} - Q_{12} Q_{22}^{-1} Q_{12}^T\right) \boldsymbol{x}_1$$

thus, the original problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \boldsymbol{x}_1^T \left(Q_{11} - Q_{12} Q_{22}^{-1} Q_{12}^T\right) \boldsymbol{x}_1 \\ \text{subject to} & g_i(\boldsymbol{x}_1) \leq 0, \quad i = 1, \ldots, m \end{array}$$

Outline

- terminology
- problem transformations
- **solving optimization problems**
- control example

Optimization methods

after constructing the mathematical model, a suitable solution method is applied to find the best decision

- an optimization *algorithm* is a set of calculations and rules that are followed to find a solution or an approximate solution to an optimization problem
- a solution method for a class of optimization problems is an algorithm that computes a solution of the problem (to some given accuracy), given a particular problem from the class, *i.e.*, an instance of the problem

Iterative algorithms

an iterative algorithm F uses the current estimate $\boldsymbol{x}^{(k)}$ at time k and the functions $f(\boldsymbol{x}), g_i(\boldsymbol{x}), h_j(\boldsymbol{x})$ to generate a new estimate $\boldsymbol{x}^{(k+1)}$ that better estimates the solution: $\boldsymbol{x}^{(k+1)} = F(\boldsymbol{x}^{(k)})$

- moving from $\boldsymbol{x}^{(k)}$ to $\boldsymbol{x}^{(k+1)}$ is called an *iteration* of the algorithm
- the algorithm stops when a good estimate of a solution is reached or until *convergence* where $\boldsymbol{x}^{(T)} = \boldsymbol{x}^{(T+1)}$ from some $T \geq 0$

Algorithm General iterative algorithm

---

given a starting point $\boldsymbol{x}^{(0)}$, a solution accuracy scalar ϵ, and error criteria

for $k \geq 1$

1. determine a search direction $\boldsymbol{d}^{(k)}$
2. determine a step-size α_k that leads to an improved estimate

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$$

3. **if** error criteria is met (*e.g.*, $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| \leq \epsilon$ or $|f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)})| \leq \epsilon$), stop and output $\boldsymbol{x}^{(k+1)}$

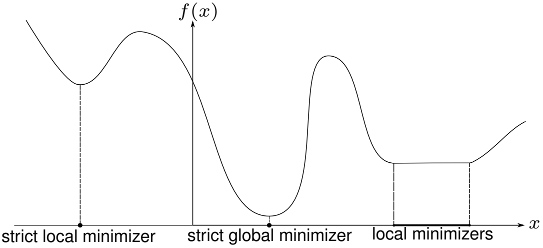
Local minimum point

$$\underset{\boldsymbol{x}\in\mathcal{X}}{\text{minimize}} \quad f(\boldsymbol{x})$$

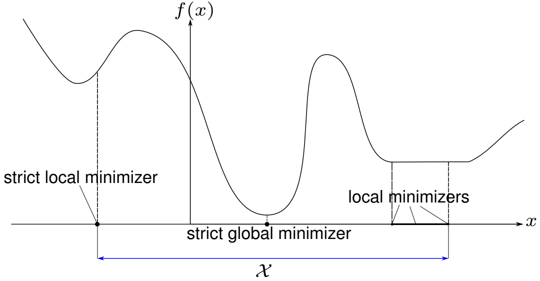
a point $\boldsymbol{x}^o \in \mathcal{X}$ is a *local minimizer* or *local minimum point* (locally optimal) if there exists a scalar $r > 0$ such that:

$$f(\boldsymbol{x}^o) \leq f(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{X} \quad \text{and} \quad \|\boldsymbol{x} - \boldsymbol{x}^o\| \leq r$$

- if $f(\boldsymbol{x}^o) < f(\boldsymbol{x})$, then the point $\boldsymbol{x}^o$ is called a *strict local minimizer*
- a point $\boldsymbol{x}^\star \in \mathcal{X}$ is a *global minimizer* or *global minimum point* (optimal) if $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$
- the term 'globally optimal' is sometimes used for 'optimal' to distinguish between 'locally optimal' and 'optimal'



unconstrained case $\mathcal{X} = \mathbb{R}^n$



constrained case $\boldsymbol{x} \in \mathcal{X}$

Local maximum point

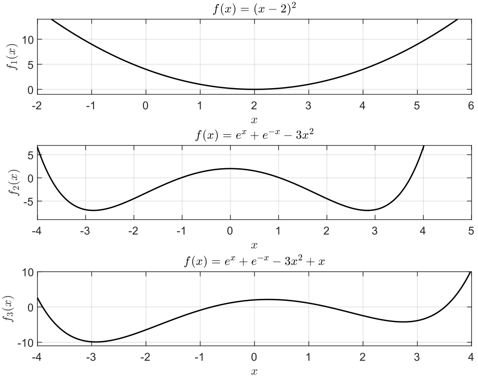
$$\underset{\boldsymbol{x} \in \mathcal{X}}{\text{maximize}} \quad f(\boldsymbol{x})$$

for maximization problems, a point $\boldsymbol{x}^o \in \mathcal{X}$ is called a *local maximizer* or *local maximum point* if (locally optimal) there exists a scalar $r > 0$ such that

$$f(\boldsymbol{x}^o) \geq f(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{X} \quad \text{and} \quad \|\boldsymbol{x} - \boldsymbol{x}^o\| \leq r$$

- if $f(\boldsymbol{x}^o) > f(\boldsymbol{x})$, then the point $\boldsymbol{x}^o$ is said to be a *strict local maximizer*
- a point $\boldsymbol{x}^\star \in \mathcal{X}$ is a *global maximizer* if $f(\boldsymbol{x}^\star) \geq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$
- a point is a local (global) maximum point of f if it is a local (global) minimum point of $-f$

Example 4.9



- $(x-2)^2$: optimal value of $\min f(x) = 0$; global minimizer $x^\star = 2$
- $e^x + e^{-x} - 3x^2$: optimal value -7.02; two global minima: $x^\star = \pm 2.84$
- $e^x + e^{-x} - 3x^2 + x$ optimal value of -9.9; global minimizer $x^\star = -2.92$; local minimizer located at $x = 2.74$

Nonlinear optimization methods

Local optimization methods

- find a locally optimal solution
- fast, can handle large-scale problems, and are widely applicable
- local optimization can be used to improve the performance of an engineering design obtained by manual, or other, design methods

Global optimization methods

- true global solution of the optimization problem is found
- difficult to find in general; even small problems, with a few tens of variables, can take a very long time (*e.g.*, hours or days) to solve
- many global optimization methods seek the global optimum by finding local solutions to a sequence of approximate subproblems

Efficiently solvable problem classes

(linear) Least squares

$$\text{minimize} \quad \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right)^2$$

where the coefficients a_{ij}, b_i are given constants

- reliable and efficient algorithms and software
- least-squares problems are easy to recognize
- many applications can be formulated as least-squares problems such as data-fitting and linear estimation

Linear program (optimization)

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m \\ & \sum_{j=1}^{n} g_{ij} x_j = h_i, \quad i = 1, \ldots, p, \end{array}$$

where the coefficients $c_j, a_{ij}, g_{ij}, h_i, b_i$ are predefined constants

- there exist robust and efficient algorithms and software for solving LPs
- LPs isn't as immediately recognizable as that of least-squares problems
- common techniques are available to transform various problems into the format of linear programs

Convex optimization

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) = g_0(\boldsymbol{x}) \\ \text{subject to} & g_i(\boldsymbol{x}) \le 0, \quad i = 1, \ldots, m, \\ & \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \ldots, p \end{array}$$

where the coefficients a_{ij}, b_i are known

- the objective and constraints functions are *convex*:

$$g_i(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \le \theta g_i(\boldsymbol{x}) + (1-\theta) g_i(\boldsymbol{y}), \quad 0 \le \theta \le 1$$

- problems with nonconvex objective or constraints are commonly referred to as *nonconvex optimization problems*

convex optimization

- include least-squares problems and linear programs as special cases
- has tons of applications
- reliable and efficient algorithms
- difficult to recognize
- many tricks can be used to transform nonconvex problems into convex form
- basis for several heuristics for solving nonconvex problems

Outline

- terminology
- problem transformations
- solving optimization problems
- **control example**

Dynamical system

a nonlinear dynamical system has the form

$$\boldsymbol{x}_{k+1} = h(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad k = 0, 1, \ldots, K$$

- $\boldsymbol{x}_k \in \mathbb{R}^n$ is the *state vector* at instant k
- $\boldsymbol{u}_k \in \mathbb{R}^m$ is the *input* or *control* at instant k
- the function $h : \mathbb{R}^{n+m} \to \mathbb{R}^n$ describes what is the next state as a function of the current state and input (evolution of the system)
- examples: vehicle dynamics, robots, chemical plants evolution...
- in optimal control, the goal is to choose the inputs $\boldsymbol{u}_0, \boldsymbol{u}_1 \ldots, \boldsymbol{u}_{K-1}$ to achieve some goal for the state and input trajectories

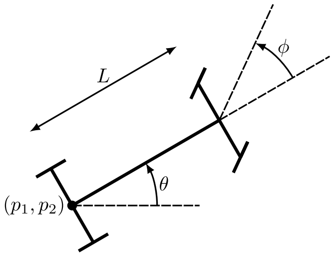
Optimal control problem

in many practical problems, the initial state $\boldsymbol{x}_0 = \boldsymbol{x}_{\text{initial}}$ is known and we want to reach a desired final state $\boldsymbol{x}_{K+1} = \boldsymbol{x}_{\text{final}}$ while minimizing some objective:

$$\begin{array}{ll} \text{minimize} & \sum_{k=0}^{K} f(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \text{subject to} & \boldsymbol{x}_1 = h(\boldsymbol{x}_{\text{initial}}, \boldsymbol{u}_0) \\ & \boldsymbol{x}_{k+1} = h(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad k = 1, \ldots, K-1 \\ & \boldsymbol{x}_{\text{final}} = h(\boldsymbol{x}_K, \boldsymbol{u}_K) \\ & g_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq 0, \quad k = 0, \ldots, K \end{array}$$

- variables $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_K$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$
- $f : \mathbb{R}^{n+m} \to \mathbb{R}$ represents a certain cost (*e.g.*, fuel consumption, time)
- $g_k : \mathbb{R}^{n+m} \to \mathbb{R}$ are some constraints functions

Car control example



$$
\begin{aligned}
\frac{dp_1}{dt}(t) &= s(t)\cos\theta(t) \\
\frac{dp_2}{dt}(t) &= s(t)\sin\theta(t) \\
\frac{d\theta}{dt}(t) &= (s(t)/L)\tan\phi(t)
\end{aligned}
$$

- wheelbase (length) L
- position $p = (p_1, p_2)$; orientation (angle) θ
- steering angle ϕ, and speed s
- we can control the speed s and the steering angle ϕ

Goal: move the car over some time period from a given initial position and orientation to a specified final position and orientation while ensuring that the input is small with little variations

Discretized car dynamics

$$
\begin{aligned}
p_1(t+\tau) &\approx p_1(t) + \tau s(t)\cos\theta(t) \\
p_2(t+\tau) &\approx p_2(t) + \tau s(t)\sin\theta(t \\
\theta(t+\tau) &\approx \theta(t) + \tau(s(t)/L)\tan\phi(t)
\end{aligned}
$$

- τ is a small time interval
- letting the state and input vectors be $\boldsymbol{x}_k = (p_1(k\tau), p_2(k\tau), \theta(k\tau))$ and $\boldsymbol{u}_k = (s(k\tau), \phi(k\tau))$, we have

$$\boldsymbol{x}_{k+1} = h\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$$

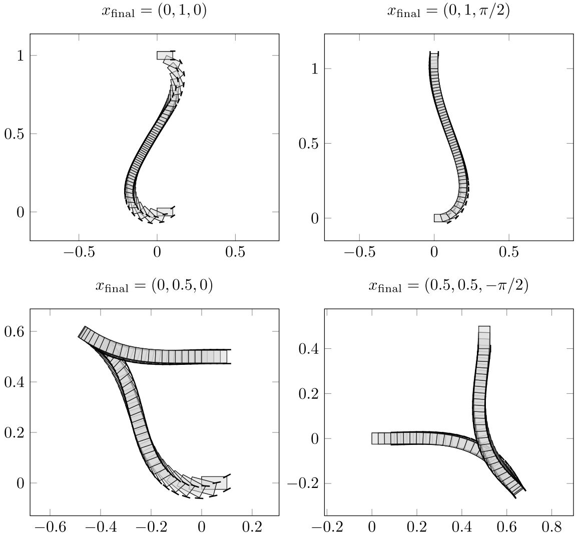
with

$$h\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) = \boldsymbol{x}_k + \tau\left(\boldsymbol{u}_k\right)_1 \left[\begin{array}{c} \cos\left(\boldsymbol{x}_k\right)_3 \\ \sin\left(\boldsymbol{x}_k\right)_3 \\ \left(\tan\left(\boldsymbol{u}_k\right)_2\right)/L \end{array}\right]$$

Problem formulation

$$\begin{array}{ll} \text{minimize} & \sum_{k=0}^{K} \|\boldsymbol{u}_k\|^2 + \rho \sum_{k=0}^{K-1} \|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\|^2 \\ \text{subject to} & \boldsymbol{x}_1 = h(\mathbf{0}, \boldsymbol{u}_0) \\ & \boldsymbol{x}_{k+1} = h(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad k = 1, \ldots, K-1 \\ & \boldsymbol{x}_{\text{final}} = h(\boldsymbol{x}_K, \boldsymbol{u}_K) \end{array}$$

- variables $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_N$, and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
- the initial state is assumed to be zero
- the objective ensures the input is small with little variation
- $\rho > 0$ is an input variation trade-off parameter



solution trajectories with different final states for $L = 0.1$, $K = 49$, $\tau = 0.1$, $\rho = 10$; the outline of the car shows the position $(p_1(k\tau); p_2(k\tau))$, orientation $\theta(k\tau)$, and the steering angle $\phi(k\tau)$ at time kh

References and further readings

- Stephen Boyd and Lieven Vandenberghe. *Convex Optimization,* Cambridge University Press, 2004, chapter 4.1.
- Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares,* Cambridge University Press, 2018 (ch 19.4, car control example).