Summary Optimiz

Adrian Willi

October 3, 2022

Chapter 1

Introduction

1.1 Optimization

1.1.1 Quantitative vs. Qualitative

Quantitative analysis and optimization (only this is covered in this course) is based on quantifiable information and knowledge like measurable data and mathematical models. Qualitative analysis and optimization is based on non-quantifiable information and knowledge like ”informal” facts and verbal descriptions of processes and procedures.

1.1.2 Continuous optimization

- Infinite number of solutions represented by continuous variables
- Graph of the objective function is arbitrary ”landscape” (i.e. nonlinear)
- Main part of the theory: **local optimization**
- Methods mainly based on differentiability information (1st and 2nd derivatives)
- Very difficult in case of non-differentiable or even non-continuous functions
- Constrained optimization (with constraints) more difficult than optimization without constraints
- **Global optimization**: mostly ”stochastic search”

1.1.3 Discrete optimization

- Number of solutions is finite (or countable): represented by integer variables
- There exists a trivial and finite algorithm: Enumeration
- Major part can be formulated as discrete problems
- Goal: to solve (exactly or approximately) a real life problem efficiently (i.e. in ”reasonable” time)
- Question: What is a good (i.e. efficient) algorithm for a given problem?
- Development of Complexity Theory (see runtime of algorithms)

1.1.4 Runtime of Algorithms

Obviously the size of the problem instance matters. But instead of measuring time, we count the number of elementary operations to be executed on a computer.

Worst-case analysis: Estimate number of elementary operations needed to solve the most difficult instance of a given size n. We only want to estimate the **order of magnitude (order of growth)** of runtime function f(n), i.e. approximative behaviour when n becomes large. Also known as Big-O notation.
Example: $7n^3 + 4n^2 + 3$ is $O(n^3)$

Polynomial Algorithms, "Good" and "Bad"

An algorithm is said to be polynomial if its runtime is $O(n^k)$ for some fixed positive number k.

"Good" vs. "Bad" Algorithms:

- Good: Polynomial runtime (considered as "effective", "tractable")
- Bad: Non-polynomial runtime (i.g. exponential, considered "intractable")

"Good" vs. "Bad" Problems:

- Good: Polynomial algorithm is known
- Bad: No polynomial algorithm is known (i.g. only exponential alg. known)

Chapter 2

Mathematical Models

2.1 Descriptive Models

Also called Evaluation models. Question: **"What if?"**. Given an alternative, calculate the resulting consequences.

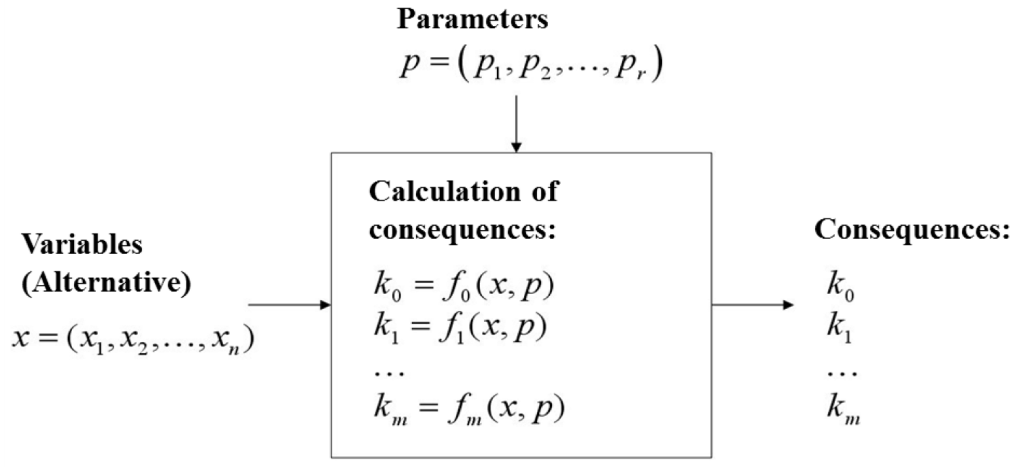


Figure 2.1: Descriptive model

How does it work?

- User **specifies** the alternative x
- Model calculates consequences $f_i(x, p)$.

Functions $f_i(\mathbf{x}, \mathbf{p})$ can be given **explicitly** or **implicitly** ($\rightarrow$ algorithm)

Example: Formulation of a descriptive model

Sets:

I Set of locations, $I = 1, 2, \ldots, 20$

Parameters:

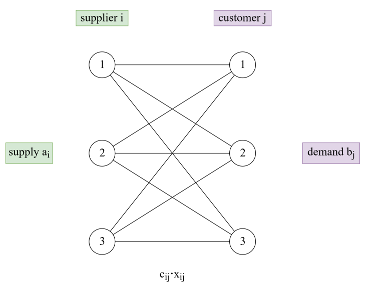
a_i Current stock of vehicles in branch i, $i \in I$

b_j Needed stock of vehicles in branch j, $j \in J$

c_{ij} Travelling distance (in km) from branch i to branch $j; i,\, j \in I$

Variables:

x_{ij} Number of transferred vehicles from branch i to branch $j; i,\, j \in I$



Consequences:

k_0	Travelling distance (in km) of all transfers
k_i^{Out}	Number of vehicles leaving branch i, $i \in I$
k_j^{In}	Number of vehicles arriving at branch j, $j \in I$

Model:

$$k_0 = \sum_{i \in I} \sum_{j \in I} c_{ij} \cdot x_{ij}$$
$$k_i^{Out} = \sum_{j \in I} x_{ij} \qquad i \in I$$
$$k_j^{In} = \sum_{i \in I} x_{ij} \qquad j \in I$$

2.2 Optimization Models

Also called Prescriptive models. Question: **"What's Best?"**. Among all **feasible** alternatives an **optimal** alternative is calculated. Set of all feasible solutions: **solution space, feasible set**

- defined by **constraints**, i.e. conditions for consequences (satisfaction)
- typically inequalities and equations

consequences to be optimized: **objective function**.

Functionality:

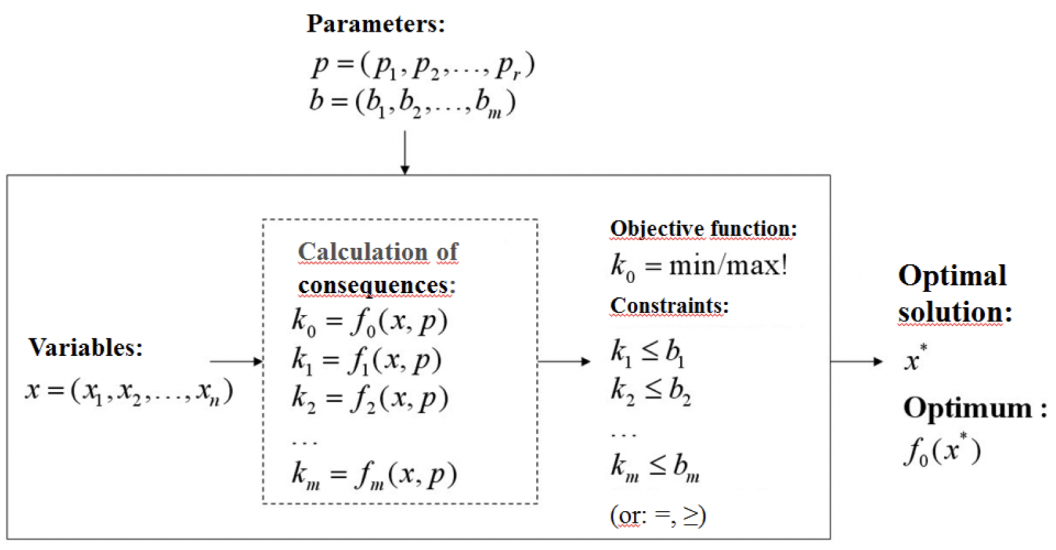


Figure 2.2: Optimization model

- No alternatives need to be specified

  feasible alternatives given by description feasible set

- Model itself provides no solution: **optimization algorithm** is necessary!

Example: Formulation of an optimization model

Sets:

I Set of locations, $I = 1, 2, \ldots, 20$

Parameters:

a_i Current stock of vehicles in branch i, $i \in I$

b_j Needed stock of vehicles in branch j, $j \in J$

c_{ij} Travelling distance (in km) from branch i to branch $j; i, j \in I$

Variables:

x_{ij} Number of transferred vehicles from branch i to branch $j; i, j \in I$

Constraints:

$$\begin{aligned} \sum_{j \in I} x_{ij} &\leq a_i \qquad i \in I \\ \sum_{i \in I} x_{ij} &\geq b_j \qquad j \in I \\ x_{ij} &\geq 0, x_{ij} \in \mathbb{Z}, \qquad i \in I, j \in I \end{aligned}$$

Objective Function:

$$\min \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij}$$

2.2.1 Notice

- Optimization model includes the mathematical definition of the feasible set
  - No alternatives need to be specified
  - Information about "construction" of alternatives integrated in model
- Optimization model includes no methods for finding optimal solutions
  - **Optimization algorithms** are needed for solving a model
  - Many different algorithms depending on the model
- Theory of optimization models and algorithms: **Operations Research**

2.3 Statement of the Model

- **General optimization problem:**

  $\prod : \max\{f(\mathbf{x}) : \mathbf{x} \in S\}$, where $S \subseteq \mathbb{R}^n$
- decision variables (Entscheidungsvariablen): $\mathbf{x} = (x_1, x_2, ..., x_n)^T \in \mathbb{R}^n$
- (feasible) solution (zulässige Lösung): $\mathbf{x} \in S$
- feasible set (Lösungsmenge): $S \subseteq \mathbb{R}^n$
- objective function (Zielfunktion): $f : S \to \mathbb{R}$

- **optimal solution:**

  $\mathbf{x}^* \in S$ such that $f(\mathbf{x}^*) \geq f(\mathbf{x})$ for all $\mathbf{x} \in S$

  This means basically, that the objective value of the optimal solution $x^* \in S$ can not be worse than the objective value of any other feasible solution ($x \in S$). Note that an optimization problem can have multiple optimal solutions, but only one optimum.

- **optimum, optimal value:** $f(\mathbf{x}^*)$

2.3.1 Conditions ensuring the Existence of an Optimal Solution

- **Feasibility:** $S \neq \emptyset$
  There must be at least one feasible solution.

  Example for infeasibility: $\max\{x_1 : 2x_1 + 4x_2 = 5, \mathbf{x} \in \mathbb{Z}^2\}$

- **Boundedness:** $\exists \omega : f(\mathbf{x}) \leq \omega$ for all $\mathbf{x} \in S$
  There exists an upper bound for the feasible solutions. For example: Tell me the highest number above 100!

  Example unbounded problem: $\max\{x_1 : 2x_1 + 4x_2 = 5, \mathbf{x} \in \mathbb{R}^2\}$

- **Closedness of S:** S is a closed set
  Example S not closed: $\max\{x_1 : x_1 < 1, x_1 \in \mathbb{R}\}$

- **Continuity of f:** f is a continuous function
  f not continuous: $\max\{f(x) : x \in \mathbb{R}, x \geq 0\}$ with

$$f(x) = \begin{cases} 3 - x, & \text{if } x > 0 \\ 2, & \text{if } x = 0 \end{cases} \tag{2.1}$$

Definition of the feasible set:

i) **Functional Constraints**: The solution space S is defined by a set of *inequalities* of the form $g_i(\mathbf{x}) \leq 0, i = 1, 2, \ldots, p (p \geq 0)$, and a set of *equalities* $h_j(\mathbf{x}) = 0, j = 1, 2, \ldots, q (q \geq 0)$:

   $S = \{\mathbf{x} \in G_1 \times \cdots \times G_n : g_i(\mathbf{x}) \leq 0, h_j(\mathbf{x}) = 0, i = 1, \ldots, p, j = 1, \ldots, q\}$
   where $G_i = \mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{R}_0, \mathbb{Q}_0, \mathbb{Z}_0, \{0, 1\}, \ldots$

ii) **Non-Functional Constraints**: The definition of the solution space S may contain "non-functional" constraints, for instance logical predicates which select the feasible elements from a given set, based on certain properties:

   $S = \{\mathbf{x} \in G_1 \times \cdots \times G_n : \mathbf{x} \text{ has certain properties}\}$

   The following example shows a non-functional definition of a solution space:

   $S = \{x \in \mathbb{Z}^n : \mathbf{x} \text{ is a permutation of the numbers } 1, 2, \ldots, n\}$

   It is theoretically possible in most cases to find, for a set defined by non-functional constraints, a corresponding formulation with functional constraints. However, finding such a formulation can be very challenging.

Maximization and Minimization

Optimization problems can be maximization or minimization problems. Every maximization problem can easily be transformed into an "equivalent" minimization problem (and vice versa), since we have

$\max\{f(\mathbf{x}) : \mathbf{x} \in S\} = -\min\{-f(\mathbf{x}) : \mathbf{x} \in S\}$

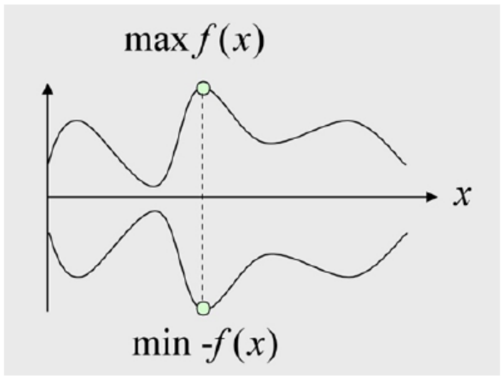


Figure 2.3: Maximization and Minimization

2.3.2 Some Notions and Concepts

Problem and Problem Instances

Problem:

$$\max\{\sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_j x_j \leq b, \mathbf{x} \in \mathbb{R}^n\}$$

Instance: Defining numerical parameters n, a, b, c

$$\max\{4x_1 + 7x_2 : 3x_1 + 5x_2 \leq 17, \mathbf{x} \in \mathbb{R}^2\}$$

Neighborhood Notions

Let $S \subseteq \mathbb{R}^n S$ be an arbitrary set of "points" in $\mathbb{R}^n$, and $P(S) = \{S' : S' \subseteq\}$ the corresponding power set, i.e. the set of all subsets of S. A *neighborhood* defined on S is a (set-valued) function of the form $N : S \to P(S)$ that assigns to each point $\mathbf{x} \in S$ a set of neighboring points $N(\mathbf{x})$, assuming always $\mathbf{x} \in N(x)$, i.e. $\mathbf{x}$ is a neighbor of itself. $N(\mathbf{x}$ is called a neighborhood of $\mathbf{x}$.

- Neighborhood: $N : S \to P(S)$ where $S \subseteq \mathbb{R}^2$
  $\mathbf{x} \mapsto N(\mathbf{x})$
- Neighborhood solutions: $N(\mathbf{x}) \subseteq S$ where $\mathbf{x} \in N(\mathbf{x})$

Euclidean Neighborhood, ε-neighborhood

A special neighborhood in $\mathbb{R}^n$ is the so-called *Euclidean Neighborhood*, or $\varepsilon - neighborhood$.

- $N_\varepsilon(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y} - \mathbf{x}\| < \varepsilon\}$ where $\varepsilon > 0$
- Euclidean norm: $\|\mathbf{x}\| = \sqrt{x_1^2 + \cdots + x_n^2}$

In $\mathbb{R}^2$ (or $\mathbb{R}^3$), the neighborhood $N_\varepsilon(\mathbf{x})$ corresponds to the set of all points $\mathbf{y}$ located inside a circle (or ball) with center $\mathbf{x}$ and radius $\varepsilon > 0$, with "boundary" points excluded. An illustration for $\mathbb{R}^2$ is given in Figure 2.4.

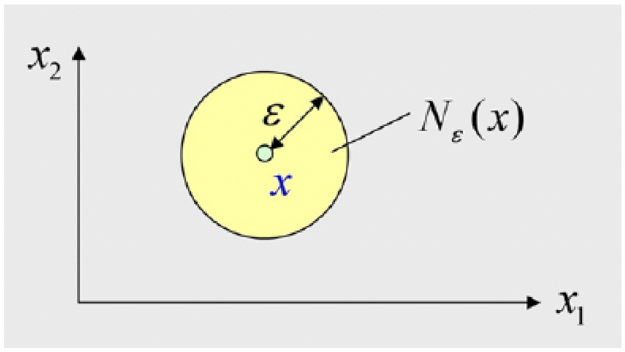


Figure 2.4: Euclidean Neighborhood, ε-neighborhood

Notions: consider $S \subseteq \mathbb{R}^n$:

- interior point $\mathbf{x}$: $N_\varepsilon(\mathbf{x}) \subseteq S$ for a suitable $\varepsilon > 0$
- boundary point $\mathbf{x}$: $N_\varepsilon(\mathbf{x}) \cap S \neq \emptyset$ and $N_\varepsilon(\mathbf{x}) \cap (\mathbb{R}^n - S) \neq \emptyset$ for all $\varepsilon > 0$
- S closed: all boundary points of S belong to S
- S open: all $\mathbf{x} \in S$ are interior points of S
- S bounded: $S \subseteq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\}$ for suitable $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$

Theorem 1. ***Extreme Value Theorem of Weierstrass***
Let $S \subseteq \mathbb{R}^n$ be a non-empty, closed, bounded set and $f : S \to \mathbb{R}$ be continuous.
Then $\max\{f(\boldsymbol{x}) : \boldsymbol{x} \in S\}$ has a finite optimum,
i.e. $\exists \mathbf{x}^* \in S$ such that $f(\mathbf{x}^*) = \max\{f(\mathbf{x} : \mathbf{x} \in S\}$.

Example: Combinatorial neighborhood
$N(\mathbf{x}) = \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{y}$ differs form $\mathbf{x}$ in at most one component$\}$

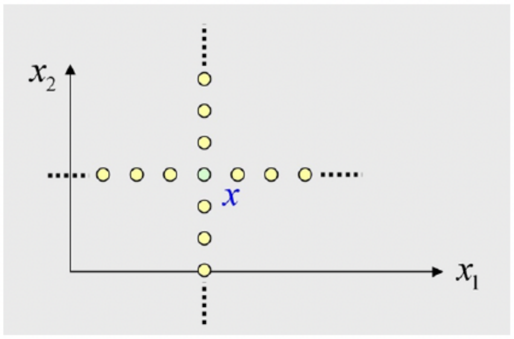


Figure 2.5: Combinatorial neighborhood

Global and Local Optima

Consider an optimization problem of the form $\prod$: $\max\{f(\mathbf{x}) : \mathbf{x} \in S\}$ with $S \subseteq \mathbb{R}^n$. As already mentioned, $\mathbf{x}^* \in S$ is called an optimal solution of $\prod$ if

- global optimal solution. $\mathbf{x}^* : f(\mathbf{x}^*) \geq f(\mathbf{x})$ for all $\mathbf{x} \in S$ (in case of maximization)

$\mathbf{x}^*$ is also called a *globally optimal solution*, and the optimum $f(\mathbf{x}^*)$ is called the *global optimum*.

Besides the global optimum of a problem, there may exist so-called *local optima* which also play an important role in optimization.

- local optimal solution $\mathbf{x}^* : \exists N(\mathbf{x}^*)$ such that $f(\mathbf{x}^*) \geq f(\mathbf{x})$ for all $\mathbf{x} \in N(\mathbf{x}^*) \cap S$.

Expressed in words, a solution is locally optimal if it is as least as good as all its neighbors in a certain neighborhood. Note that every globally optimal solution is also a locally optimal solution **(but not conversely)**.

Local optimal solution is referring to Euclidean neighborhood $N_\varepsilon(\mathbf{x})$ or to different neighborhoods. For different neighborhoods different local optimal solutions are possible.

Figure 2.6 shows an example in $\mathbb{R}^1$. Note that in $\mathbb{R}^1$, the ε-neighborhoods to be considered for checking local optimality of a solution $x \in \mathbb{R}^1$ correspond to the open intervals $]x - \varepsilon, x + \varepsilon[$.

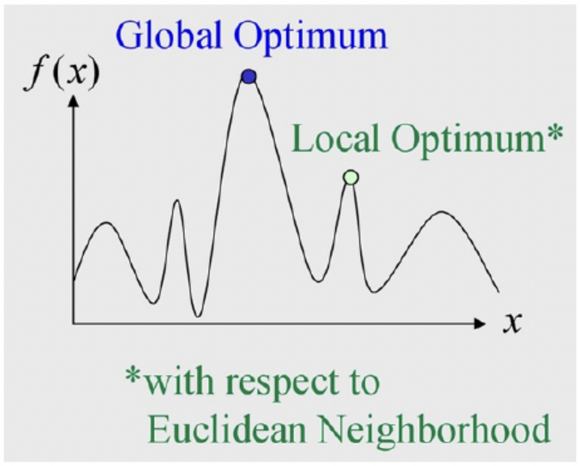


Figure 2.6: Global and Local Optima

Level Sets

Levelsets of functions play an important role in analysis and optimization, notably for the graphical representation of 2- and 3-dimensional functions. Recall that the graph H of a function $f : S \rightarrow$ with $S \subseteq \mathbb{R}^n$ is the $n + 1$ dimensional subset $H \subseteq \mathbb{R}^{n+1}$ defined by

- $H = \{(\mathbf{x}, f(\mathbf{x})) : \mathbf{x} \in S\}$
- Level set for level α is:
  $L_\alpha = \{\mathbf{x} \in S : f(\mathbf{x}) = \alpha\}$.

  When the number of variables is two, a level set is generally a curve and called a level curve or contour line. For n = 3, level sets are typically ”surfaces” in the 3-dimensional space.

  By means of level sets, graphs of 2- and 3-dimensional functions (which are 3- and 4-dimensional sets, respectively) can be represented in lower dimension as 2- and 3- dimensional images, respectively. An example is given in Figure2.7.

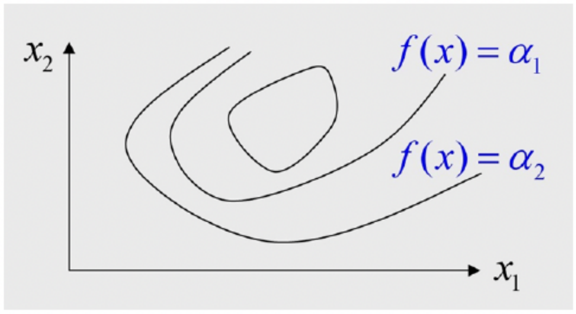


Figure 2.7: Level set

- **Special case: linear functions**
  If $f : \mathbb{R}^2 \to \mathbb{R}$ is a linear function, i.e. if $f(\mathbf{x}) = c_1x_1 + c_2X_2$ for some $\mathbf{c} \in \mathbb{R}^2$, the level sets are parallel lines in $\mathbb{R}^2$, and the vector $\mathbf{c}$ (the gradient of f) is orthogonal to these lines. The vector $\mathbf{c}$ points in the direction where the level α (i.e. function value) increases if level lines are "shifted in parallel" in this direction. See Figure 2.8 for an example.

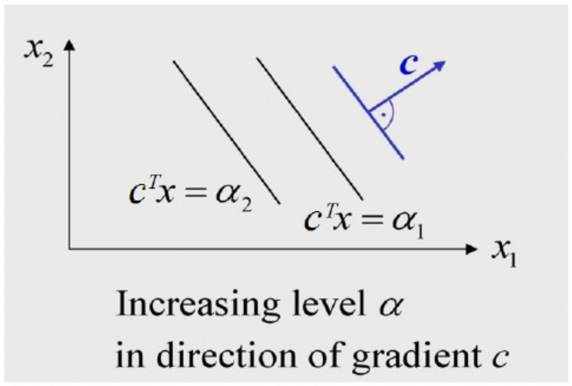


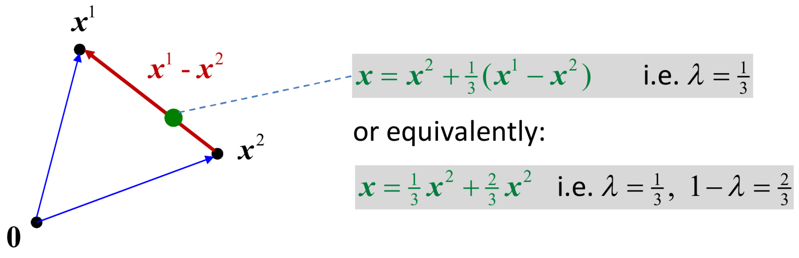
Figure 2.8: Special case: linear functions

2.3.3 Convex Optimization

Convex combination of two points $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^n$:

$$\begin{aligned} \mathbf{x} &= \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 && \text{for some } \lambda \in \mathbb{R} \text{ with } 0 \le \lambda \le 1 \\ \mathbf{x} &= \mathbf{x}^2 + \lambda(\mathbf{x}^1 - \mathbf{x}^2) && \text{for some } \lambda \in \mathbb{R} \text{ with } 0 \le \lambda \le 1 \end{aligned}$$

Geometrically this can be understood as a convex combination of $\mathbf{x}^1$ and $\mathbf{x}^2$ that lies on the line between $\mathbf{x}^1$ and $\mathbf{x}^2$.


x^1
$x^1 - x^2$
x^2
0
$x = x^2 + \frac{1}{3}(x^1 - x^2)$ i.e. $\lambda = \frac{1}{3}$
or equivalently:
$x = \frac{1}{3}x^2 + \frac{2}{3}x^2$ i.e. $\lambda = \frac{1}{3},\ 1-\lambda = \frac{2}{3}$

A **convex combination of multiple points** $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k \in \mathbb{R}^n$ takes the form:

$$\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}^i \qquad \text{for some } \boldsymbol{\lambda} \in \mathbb{R}^k \text{ with } 0 \leq \lambda \leq 1 \text{ and } \sum_{i=1}^{k} \lambda_i = 1$$

Expressed in words: A convex combination is a linear combination with non-negative coefficients that sum up to 1.

Convex set $S \subseteq \mathbb{R}^n$:

$$\lambda \mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 \in S \quad \forall \mathbf{x}^1, \mathbf{x}^2 \in S \text{ and all } \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1$$

Expressed in words: A set S is convex if the line between two arbitrary points of S is entirely contained in S.

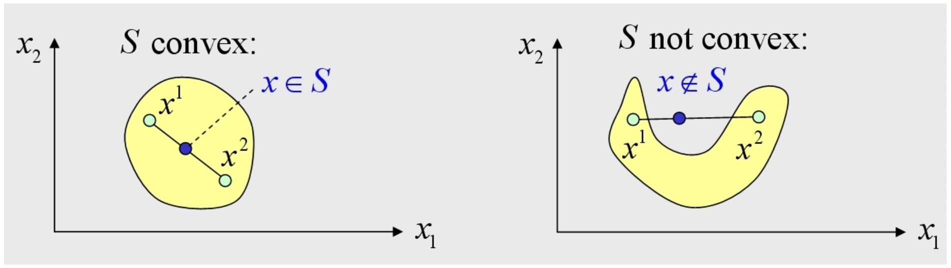


Figure 2.9: Illustration of convex and non-convex sets

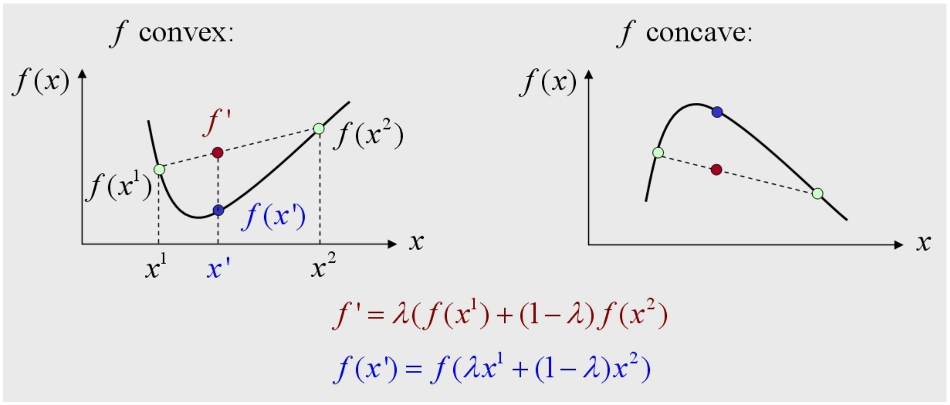
Theorem 2. *Any intersection of convex sets is a convex set.*

Convex function $f : S \to \mathbb{R}$, where S convex:

$$f(\lambda \mathbf{x}^1 + (1-\lambda)\mathbf{x}^2) \leq \lambda f(\mathbf{x}^1) + (1-\lambda) f(\mathbf{x}^2) \qquad \forall \mathbf{x}^1, \mathbf{x}^2 \in S, \lambda \in [0,1]$$

Concave function $f : S \to \mathbb{R}$, where S convex:

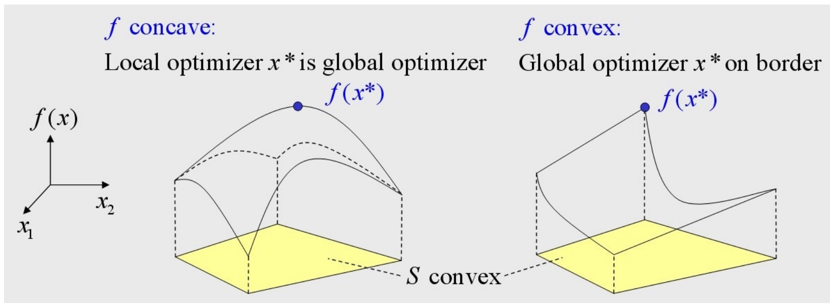
$$f(\lambda \mathbf{x}^1 + (1-\lambda)\mathbf{x}^2) \geq \lambda f(\mathbf{x}^1) + (1-\lambda) f(\mathbf{x}^2) \qquad \forall \mathbf{x}^1, \mathbf{x}^2 \in S, \lambda \in [0,1]$$



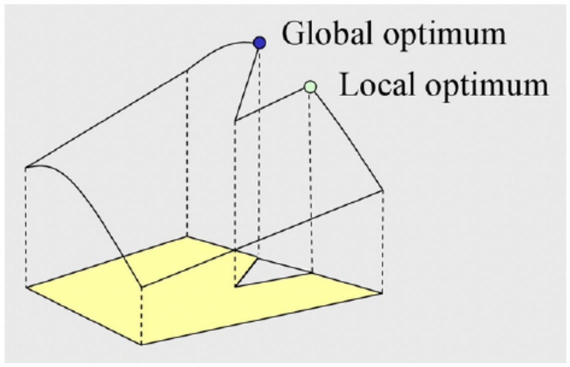
f convex:
$f(x)$
f'
$f(x^2)$
$f(x^1)$
$f(x')$
x^1
x'
x^2
x
f concave:
$f(x)$
x
$f' = \lambda(f(x^1) + (1-\lambda) f(x^2)$
$f(x') = f(\lambda x^1 + (1-\lambda) x^2)$

- **Linear function**: convex and concave
- Convex optimization problem: $\max\{f(\mathbf{x}) : \mathbf{x} \in S\}$ with f **convex** and let S be **convex**
- "Convex" optimization problem as well: $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$ with f **concave** and S **convex**

Theorem 3. *For a convex optimization problem any **local optimum** is a **global optimum**.*


f concave:
Local optimizer x^* is global optimizer
$f(x^*)$
f convex:
Global optimizer x^* on border
$f(x^*)$
$f(x)$
x_2
x_1
S convex

Theorem 4. *Consider the problem $\max\{f(\mathbf{x}) : \mathbf{x} \in S\}$ with f **convex** and let S be **convex**, closed and bounded (alternatively consider $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$ with f **concave**). Further, let f not be constant in S. Then any **local optimum** is a **boundary point** of S.*



Convex programming :

$$\max\{f(\mathbf{x}) : g_i(\mathbf{x}) \le b_i, i \in I, \mathbf{x} \in \mathbb{R}^n\}$$

where f concave and $g_i, i \in I$ are all convex.

Theorem 5. $S = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \le b_i, i \in \mathbf{I}\}$ *is convex.*

2.4 Types of optimization problems and methods

2.4.1 Optimization models:

- Constrained vs. Unconstrained
- Global vs. Local
- Differentiable vs. Non-differentiable
- Discrete vs. Continuous
- Convex vs. Non-convex
- Linear vs. Nonlinear

2.4.2 Optimization methods:

- Exact vs. Heuristic
- General vs. Problem specified

Types of heuristics:

- Constructive
- Improving heuristics
  - Local Search
  - general Meta Heuristics

Bibliography