40.002 OPTIMIZATION

An Introduction to Optimization

Michael Hoon

January 29, 2024

Contents

1 Introduction to Linear Programming

An optimization problem is defined by:

- **Decision variables**: elements under the control of the decision maker
- **A (single) objective function**: a function of the decision variables that we want to optimize, corresponding to a criterion for measuring maximize
- **Constraints**: restrictions that define which values of the decision variables are allowed.

We want to find the **minimum** or **maximum** of a function of one or many variables subject to a set of **constraints**:

$$\begin{aligned} &\min \ f(x_1, \dots x_n) \\ &\ni (x_1, \dots x_n) \in \chi \subseteq \mathbb{R}^n \end{aligned} \tag{1}$$

where the decision variables are vectors $x_1, \dots x_n$, the objective function is $f(x_1, \dots x_n)$ and the constraints are defined by the set $\chi \subseteq \mathbb{R}^n$. A vector $\mathbf{x}^*$ is called *optimal*, or a *solution* of the problem, if it has the **smallest objective value** among all vectors that satisfy the constraints.

1.1 Standard Form

A linear program is a class of optimisation problem in which the objective and all constraint functions are **linear**. For a minimisation problem,

$$\begin{aligned} &\min \ \mathbf{c}^\top \mathbf{x} \\ &\ni \mathbf{A}\mathbf{x} \geq \mathbf{b}, \text{ and } \mathbf{x} \geq 0 \end{aligned} \tag{2}$$

and for maximisation problems,

$$\begin{aligned} &\max \ \mathbf{c}^\top \mathbf{x} \\ &\ni \mathbf{A}\mathbf{x} \leq \mathbf{b}, \text{ and } \mathbf{x} \geq 0 \end{aligned} \tag{3}$$

where the decision vector is $\mathbf{x}$ (n variables), linear objective function: $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} = \sum_{i=1}^{n} c_i x_i$, and the linear constraints are $\chi = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$ (m constraints) [1]. Note that matrix $\mathbf{A}_{(m \times n)}$ is of $m \times n$ dimension.

1.1.1 Inequality Transformations

We have matrix $\mathbf{A}$ given by:

$$\mathbf{A} = \begin{pmatrix} - & \mathbf{a_1}^\top & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{a_m}^\top & - \end{pmatrix}$$

- An equality constraint $\mathbf{a_i}^\top \mathbf{x} = b_i$ is equivalent to two equality constraints $\mathbf{a_i}^\top \mathbf{x} \leq b_i$ and $\mathbf{a_i}^\top \mathbf{x} \geq b_i$
- An inequality constraint $\mathbf{a_i}^\top \mathbf{x} \leq b_i$ is equivalent to the inequality constraint $-\mathbf{a_i}^\top \mathbf{x} \geq -b_i$ (Note the negatives applied to both sides of the inequality).

[1] Note: vector inequalities are interpreted componentwise.

- Constraints such as $x_j \geq 0,\ x_j \leq 0$ can be expressed in the form $\mathbf{a_i}^\top \mathbf{x} \geq b_i$ by appropriately choosing $\mathbf{a}_i,\ b_i$.

Note that there is no simple analytic formula for the solution of a linear program, but there are a variety of effective methods for solving them, including Dantzig's simplex method, and the more recent interior-point methods. We cannot give the exact number of arithmetic operations required to solve a linear program, but we can establish rigorous bounds on the number of operations required to solve a linear program using an interior-point method (in practice, this is of the order n^2m, assuming $m \geq n$).

1.1.2 Terminology

Definition 1.1. We now introduce some terminology for geometric linear programming:

- A **linear function** $f : \mathbb{R}^n \to \mathbb{R}$ is a function of the form:

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n} a_i x_i, \quad a_i \in \mathbb{R}$$

- A **hyperplane** in $\mathbb{R}^n$ is the set of points satisfying a single linear **equation**:

$$a_1x_1 + \cdots + a_nx_n = b, \quad a_n \in \mathbb{R}$$

- A **halfspace** in $\mathbb{R}^n$ is the set of points satisfying a single linear **constraint**:

$$a_1x_1 + \cdots + a_nx_n \geq b, \quad a_n, b \in \mathbb{R}$$

  A halfspace is a **convex set**.

- An LP is **bounded** if there is some value Z such that $\mathbf{c}^\top \mathbf{x} \leq Z$.
- A **polyhedron** is a set that can be described by a finite number of halfspaces. A **polytope** is a **bounded** polyhedron. The polytope of an LP is **convex**, since it is the intersection of halfspaces (which are convex).
- An assignment of values to the decision variables is a **feasible solution** if it **satisfies all the constraints** (infeasible otherwise). The set of all feasible solutions is the **feasible region.**
- An **optimal solution** is a feasible solution that achieves the **best possible objective function value**. For a minimisation problem, x^* is optimal **iff** $\mathbf{c}^\top \mathbf{x}^\top \leq \mathbf{c}^\top \mathbf{x}$ for all feasible $\mathbf{x}$.
- We call $\mathbf{c}^\top x^*$ the **optimal objective value**.
- $\forall K \in \mathbb{R}$ we can find a feasible solution $\mathbf{x}$ such that $\mathbf{c}^\top \mathbf{x} \leq K$, then the linear program in **minimisation** form has **unbounded** cost. The optimum cost is then $-\infty$. In this case, we can find a feasible $\mathbf{x}$ and direction $\mathbf{d}$ such that $\mathbf{x} + t\mathbf{d}$ is feasible $\forall t \geq 0$ and $\mathbf{c}^\top d < 0$.

For every linear program, we know that one of the following cases must hold:

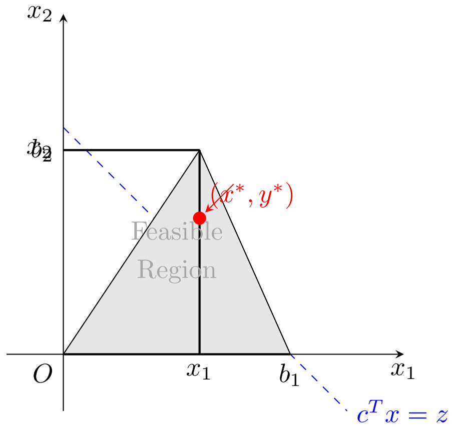
1. The LP is infeasible. There is no value of x that satisfies the constraints.
2. The LP has an optimal solution
3. The LP is unbounded.

Mathematically, this follows from the fact that if the LP is **feasible and bounded**, then it is a closed and bounded subset of $\mathbb{R}^n$ and hence has a maximum point.

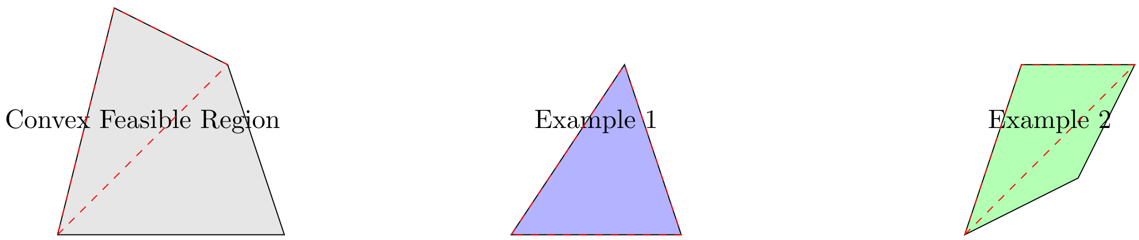
1.2 Geometric Definition

In a simple two-dimensional space with the equation $x_1 + x_2 = z$, this function can be represented by a line. The decision variables are x_1 ad x_2, and this line represents all possible combinations of x_1 and x_2 that yield the same objective value z.

Each constraint is a linear inequality, which creates a boundary in the solution space. The feasible region is now the polygon formed by the intersection of all these constraint boundaries.



The feasible region is often convex, meaning that if points A and B are inside the region, the line segment connecting A and B are also inside the region. Some examples are shown below:



1.3 Graphical Approach

Solving the LP via a graphical approach involves drawing the halfspaces defined by the constraints, as well as the iso-lines defined by the optimisation problem.

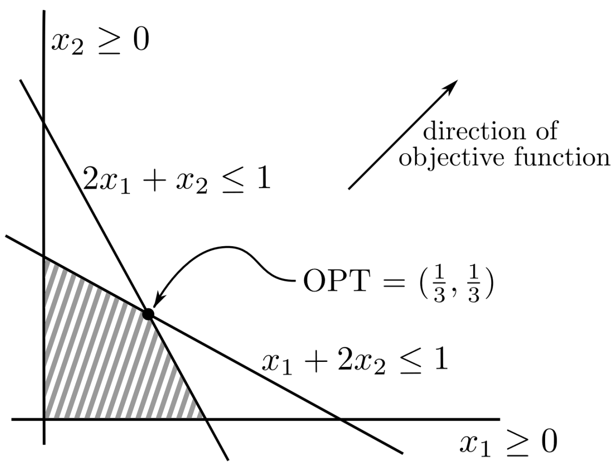


Figure 1: A linear Program in 2 dimensions

After determining the bounded-ness of the LP problem, shade the feasible region (polytope) defined by the constraints. Points within this feasible region satisfy all constraints, and is a **convex polygon**.

Note that the feasible region for an LP may be *empty, a single point, or infinite (unbounded)*. Our goal now is to find a point in the feasible region that optimises this objective. Since there are an infinite number of possible points within the polytope, we need to reduce this space.

Theorem 1.2. The maximum point for an LP is always achieved at one of the vertices of a polytope. In general, if there are n dimensions (variables), a vertex may occur wherever n (linearly independent) hyperplanes (i.e. constraints) intersect.

Recall in linear algebra that if you have n linearly independent equations and n variables, there is only one optimal solution (and if they are not linearly independent - you have infinitely many solutions). So in a system with m constraints and n variables, there are $\binom{m}{n} = O(m^n)$ vertices.

Now, to determine which of the vertices gives the maximum objective value, we can substitute the variables into the objective function and compare the final values [2]

1.3.1 Geometric Intuition

The objective function gives the optimisation direction, and the goal is to find the feasible point that is furthest in this direction.

1.4 Convexity

When optimisation is concerned, we equate "convex" with "nice", and "non-convex" with "nasty".

1.4.1 Convex Sets

Theorem 1.3. The feasible region (Polytope) of an LP is **convex**.

Intuitively, a subset $C \subseteq \mathbb{R}^n$ is convex if it is "filled in", meaning that it contains all line segments between its points (if you draw a line segment between two points of the region, the line segment itself must be in the region).

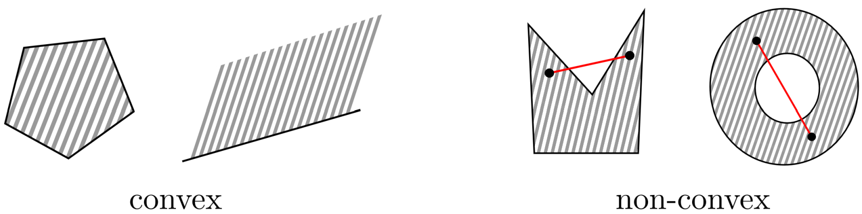


Figure 2: Examples of convex and non-convex sets

Mathematically, a set $\chi \subseteq \mathbb{R}^n$ is convex if

$$\lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in \chi, \ \forall \mathbf{x}, \mathbf{y} \in \chi, \ \forall \lambda \in [0,1]$$

As λ ranges from 0 to 1, it traces out the line segment from $\mathbf{y}$ to $\mathbf{x}$.

[2] Note that at each vertex, if the iso-line falls within the polytope, then it is not a maximum.

1.4.2 Convex Functions

We define a function $f : \mathbb{R}^n \to \mathbb{R}$ to be *convex* if and only if the region above its graph is a convex set.

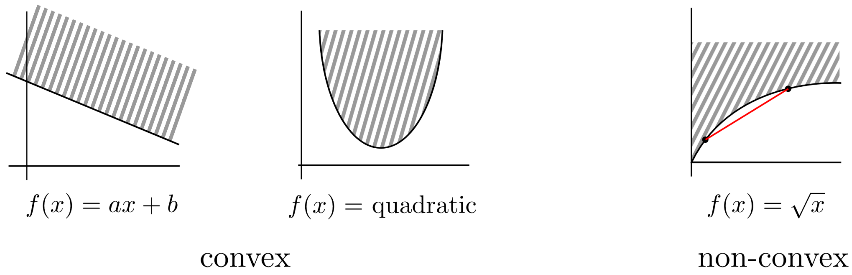


Figure 3: Examples of convex and non-convex functions

Equivalently, a convex function is one where all "chords" of its graph lie above the graph. Mathematically,

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}),\ \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,\ \forall \lambda \in [0,1] \tag{4}$$

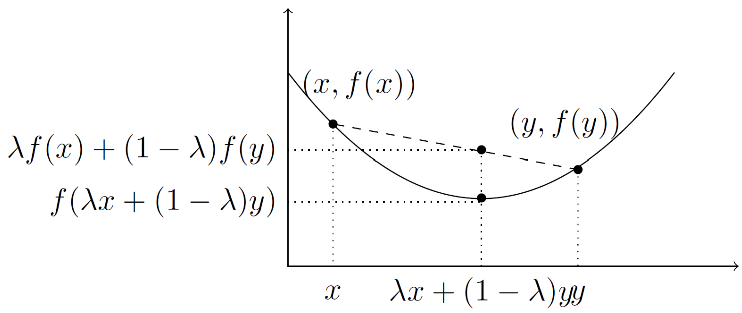


Figure 4: Visualisation of a convex function

That is, for points $\mathbf{x}$ and $\mathbf{y}$, if you take the average of $\mathbf{x}$ and $\mathbf{y}$, and then apply f, you'll get a smaller number than if you first apply f to $\mathbf{x}$ and $\mathbf{y}$, and then average the results. Likewise, a function f is **concave** if $-f$ is convex.

1.4.3 Why Convexity Helps

Consider the case where the feasible region or the objective function is not convex. With a non-convex feasible region, there can be "locally optimal" feasible points that are not globally optimal, even with a linear objective function. The same problem arises with a non-convex objective function, even when the feasible region is just the real line. When both the objective function and feasible region are convex, this cannot happen - **all local optima are also global optima** (which makes optimisation easier).

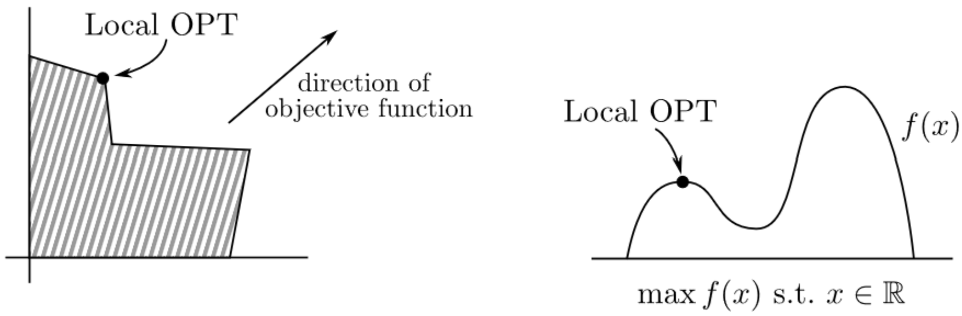


Figure 5: Non-convexity and local optima. (Left) A linear (i.e. convex) objective function with a non-convex feasible region. (Right) A non-convex objective function over a convex feasible region (the real line).

1.4.4 First-Order Characterisation

1.4.5 Second-Order Characterisation

Suppose that a function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable (i.e. the Hessian $\nabla_x^n f(x)$ is defined for all x in the domain of f). Then f is convex if and only if $D(f)$ is a convex set and its Hessian is positive semidefinite (PSD): i.e. for any $x \in D(f)$,

$$\nabla_x^2 f(x) \succeq 0$$

where $\succeq$ denotes PSD-ness. In one-dimension, this is equivalent to the condition that the second derivative $f''(x)$ always be non-negative.

The Hessian is defined as:

$$\nabla_x^2 f(x) \in \mathbb{R}^{n \times n}, \quad (\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

Theorem 1.4. We can see that f is **strictly convex** if its Hessian is **Positive Definite**, concave if it is **Negative Semidefinite**, and **strictly concave** if it is **Negative Definite**.

1.4.6 Definite Matrix & Eigenvalues

Definition 1.5. Let M be an $n \times n$ Hermitian Matrix [e](including Symmetric Matrices [a]). All eigenvalues of M are real, and their sign characterise its definite-ness:

1. M is **Positive Definite** if and only if all of its eigenvalues are **positive**.
2. M is **Positive Semi-Definite** if and only if all its eigenvalues are **non-negative**.
3. M is **Negative Definite** if and only if all of its eigenvalue are **negative**.
4. M is **Negative Semi-Definite** if and only if all if its eigenvalues are **non-positive**.
5. M is **indefinite** if and only if it has both positive and negative eigenvalues.

[e] A complex square matrix that is equal to its own conjugate transpose - $A = \overline{A^\top}$.
[a] A square matrix that is equal to its transpose.

1.4.7 Solving Convexity-Related Questions

2 Simplex Method

3 Duality

4 Sensitivity Analysis

5 Application to Game Theory

6 Robust Optimization

7 Maximum Matching

8 Network Simplex Algorithm

9 Integer Programming

10 LP Relaxation

11 Branch-and-Bound

12 Cutting Planes

13 Dynamic Programming

14 Travelling Salesman Problem