
40.002 OPTIMIZATION

An Introduction to Optimization

Michael Hoon

February 5, 2024

Contents

1	Introduction to Linear Programming	3
1.1	Standard Form	3
1.1.1	Inequality Transformations	3
1.1.2	Terminology	4
1.2	Geometric Definition	5
1.3	Graphical Approach	5
1.3.1	Geometric Intuition	6
1.4	Convexity	6
1.4.1	Convex Sets	6
1.4.2	Convex Functions	7
1.4.3	Why Convexity Helps	7
1.4.4	First-Order Characterisation	8
1.4.5	Second-Order Characterisation	8
1.4.6	Solving Convexity-Related Questions	9
1.5	Some Matrix Calculus	9
1.5.1	The Gradient	9
1.5.2	The Hessian	9
1.5.3	Definite Matrix & Eigenvalues	10
2	Simplex Method	10
2.1	The True Standard Form	10
2.1.1	Transformation Tricks	10
2.2	Active Constraints	11
2.3	Some Polyhedron Definitions	11
2.4	Finding a Basic Feasible Solution	11
2.5	The Simplex Method	12
2.5.1	Phase I	12
3	Duality	12
4	Sensitivity Analysis	12
5	Application to Game Theory	12
6	Robust Optimization	12
7	Maximum Matching	12
8	Network Simplex Algorithm	12
9	Integer Programming	12
10	LP Relaxation	12
11	Branch-and-Bound	12
12	Cutting Planes	12
13	Dynamic Programming	12
14	Travelling Salesman Problem	12

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**:

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

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,

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

and for maximisation problems,

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

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 $\chi = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \geq \mathbf{b}\}$ (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).

¹Note: 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** $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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**:

$$a_1 x_1 + \dots + a_n x_n = b, \quad a_n \in \mathbb{R}$$

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

$$a_1 x_1 + \dots + a_n x_n \geq b, \quad a_n, b \in \mathbb{R}$$

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** $\mathbf{c}^\top \mathbf{x}^* \leq \mathbf{c}^\top \mathbf{x}$ 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 $\mathbf{c}^\top \mathbf{d} < 0$.

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

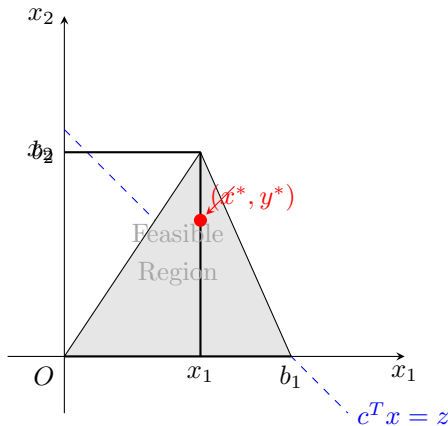
1. The LP is infeasible. There is no value of x that satisfies the constraints.
2. The LP has an optimal solution
3. The LP is unbounded.

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.

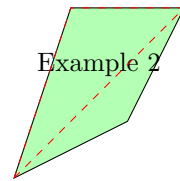
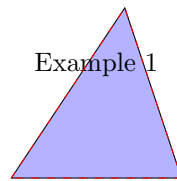
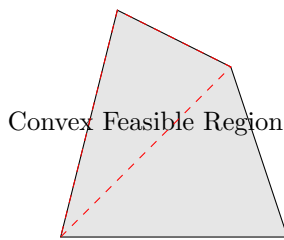
1.2 Geometric Definition

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 and 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.

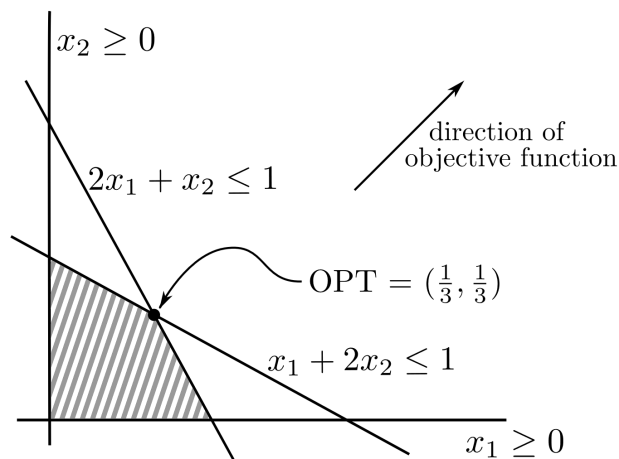


Figure 1: A linear Program in 2 dimensions

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).

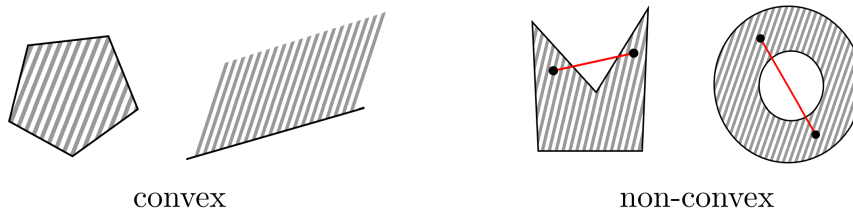


Figure 2: Examples of convex and non-convex sets

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} .

²Note 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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ to be *convex* if and only if the region above its graph is a convex set.

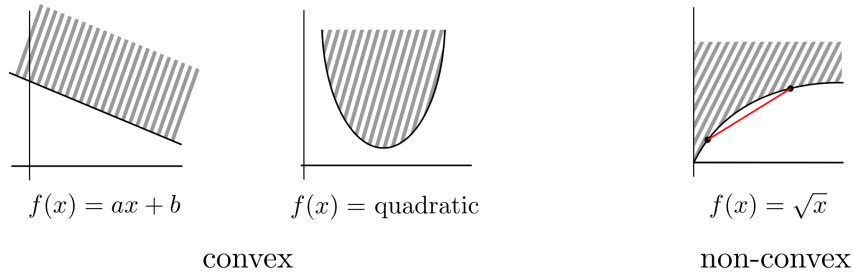


Figure 3: Examples of convex and non-convex functions

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

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

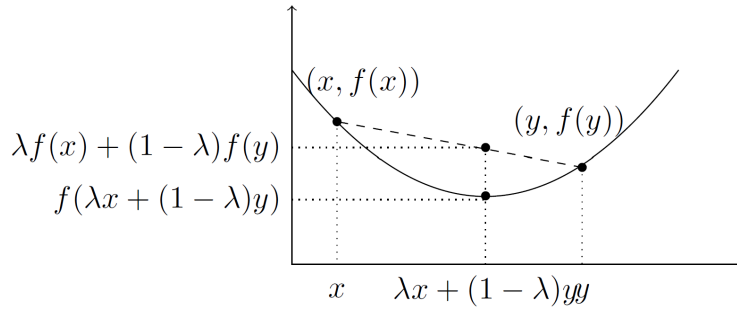


Figure 4: Visualisation of a convex function

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).

Theorem 1.4. Let χ be a convex set and $f(x)$ be a convex function. Then:

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x \in \chi \subseteq \mathbb{R}^n \end{aligned}$$

is a convex optimisation problem. A key property of such a convex optimisation problem that is a **local minimum is always the global minimum**.

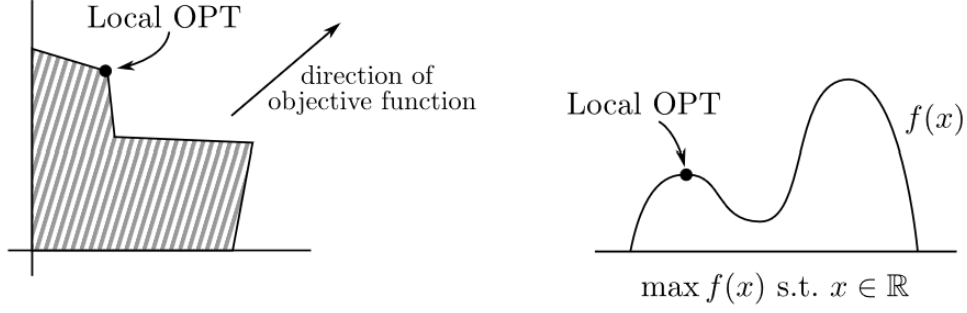


Figure 5: Non-convexity and local optima. (Left) A linear (i.e. convex) objective function with a non-convex feasible region. (Right) A non-convex objective function over a convex feasible region (the real line).

1.4.4 First-Order Characterisation

Suppose a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable³. Then f is convex if and only if $D(f)$ is a convex set and for all $x, y \in D(f)$:

$$f(y) \geq f(x) + \nabla_x f(x)^\top (y - x) \quad (5)$$

The function on the right is called the **first order approximation** to f at the point x .

Theorem 1.5. The first order condition for convexity says that f is convex if and only if the tangent line is a global underestimator of the function f . i.e. if we draw a tangent line at any point (refer to Figure ??), then every point on this line will lie below the corresponding point on f .

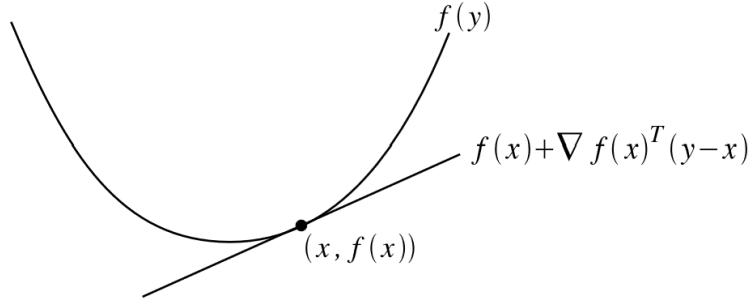


Figure 6: Illustration of first-order condition for convexity

1.4.5 Second-Order Characterisation

Suppose that a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable (i.e. the Hessian $\nabla_x^2 f(x)$ 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 $x \in D(f)$,

$$\nabla_x^2 f(x) \succeq 0 \quad (6)$$

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:

³the gradient $\nabla_x f(x)$ exists at all points x in the domain of f . It is defined as $\nabla_x f(x) \in \mathbb{R}^n$, $(\nabla_x f(x))_i = \frac{\partial f(x)}{\partial x_i}$

$$\nabla_x^2 f(x) \in \mathbb{R}^{n \times n}, \quad (\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \quad (7)$$

Theorem 1.6. 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 Solving Convexity-Related Questions

1.5 Some Matrix Calculus

1.5.1 The Gradient

Definition 1.7. Suppose that $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ is a function that takes as input a matrix A of size $m \times n$ and returns a real value. Then the **gradient** of f (with respect to $A \in \mathbb{R}^{m \times n}$) is the matrix of **partial derivatives**:

$$\nabla_A f(A) \in \mathbb{R}^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix} \quad (8)$$

Note that the gradient of a function ⁴ is only defined if the function is real-valued, i.e. if it returns a scalar value.

Some equivalent properties of the gradient from partial derivatives:

1. $\nabla_x(f(x) + g(x)) = \nabla_x f(x) + \nabla_x g(x)$
2. For $t \in \mathbb{R}$, $\nabla_x(tf(x)) = t\nabla_x f(x)$

1.5.2 The Hessian

Definition 1.8. Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function that takes a vector in \mathbb{R}^n and returns a real number. Then the **Hessian** with respect to x , written $\nabla_x^2 f(x)$ is the $n \times n$ matrix of partial derivatives:

$$\nabla_x^2 f(x) \in \mathbb{R}^{n \times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \quad (9)$$

Note that the Hessian is always symmetric, since

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}$$

The Hessian is defined only when $f(x)$ is real-valued. Note that for functions of a vector, the gradient of the function is a vector, and we cannot take the gradient of a vector. **Therefore, it is not the case that the Hessian is the Gradient of the Gradient.**

⁴Gradients are a natural extension of partial derivatives to functions of multiple variables.

1.5.3 Definite Matrix & Eigenvalues

Definition 1.9. Let M be an $n \times n$ Hermitian Matrix ^a(including Symmetric Matrices ^b). 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.

^aA complex square matrix that is equal to its own conjugate transpose: $A = \overline{A}^\top$.

^bA square matrix that is equal to its transpose.

2 Simplex Method

2.1 The True Standard Form

Previously, we know that a linear program can take either a maximisation or minimisation form, depending on the context. The constraints thus can either be inequalities or equalities. Some variables might be unrestricted in sign, while others might be restricted to be non-negative.

A linear program is said to be in *standard form* if the following hold:

1. It is a minimisation program.
2. There are only equalities (no inequalities) and
3. All variables are restricted to be non-negative.

In matrix form, we have

$$\begin{aligned} \min \quad & c^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned} \tag{10}$$

where n is the number of decision variables and m is the number of equality constraints and $\mathbf{0}$ is a vector of zeros.

2.1.1 Transformation Tricks

We have a few transformation tricks to convert any LP into the true standard form:

1. Eliminate non-positive and free variables (unrestricted in sign):

$$x_j \leq 0 \rightarrow \text{Let } \hat{x}_j = -x_j \text{ and set } \hat{x}_j \geq 0$$

$$x_j \text{ is free} \rightarrow \text{Let } x_j = x_j^+ - x_j^- \text{ and set } x_j^+, x_j^- \geq 0$$

2. If we have an inequality constraint $a_{i1}x_1 + \dots + a_{in}x_n \leq b$, then we can transform it into an equality constraint by adding a *slack* variable s_i , restricted to be non-negative:

$$a_{i1}x_1 + \dots + a_{in}x_n + s_i = b_i, \quad s_i \geq 0$$

3. Similarly, if we have an inequality constraint $a_{i_1}x_1 + \dots + a_{i_n}x_n \geq b_i$ then we can transform it into an equality constraint by adding a *surplus* variable, s_i , restricted to be non-negative:

$$a_{i_1}x_1 + \dots + a_{i_n}x_n - s_i = b_i, \quad s_i \geq 0$$

2.2 Active Constraints

A constraint is said to be **active** at a point \mathbf{x} if the constraint is satisfied at equality at that point.

A set of linear constraints are linearly independent if their coefficient vectors are linearly independent; else they are linearly dependent.

2.3 Some Polyhedron Definitions

Let $P \subseteq \mathbb{R}^n$ be a non-empty polyhedron.

1. A vector $\mathbf{x} \in P$ is an **extreme point** of P if we cannot find two vectors $\mathbf{y}, \mathbf{z} \in P$, both different from \mathbf{x} , and a scalar $\lambda \in [0, 1]$, such that $\mathbf{x} = \lambda\mathbf{y} + (1 - \lambda)\mathbf{z}$ (Geometric Definition).
2. A vector $\mathbf{x} \in P$ is a **vertex** of P if there exists some \mathbf{c} such that \mathbf{x} is the unique minimizer of the LP $\min\{\mathbf{c}^\top \mathbf{y} \mid \mathbf{y} \in P\}$ (Optimisation definition)
3. A vector $\mathbf{x} \in P$ is a **basic feasible solution** (BFS) if there exists n linearly independent constraints that are active at \mathbf{x} (including both inequality and equality constraints) (Algebraic definition).

Theorem 2.1. Let $P \subseteq \mathbb{R}^n$ be a nonempty polyhedron and let $x^* \in P$. Then:

$$x^* \text{ is an extreme point} \iff x^* \text{ is a vertex} \iff x^* \text{ is a BFS.}$$

Notably, these 3 are the same definitions but in different form.

2.4 Finding a Basic Feasible Solution

We can "eyeball" a BFS by the following steps:

Consider an LP where:

1. LP is of standard form
2. The right hand side vector \mathbf{b} is non-negative
3. The constraint matrix contains the identity matrix I as a **submatrix** (or a permutation of the identity matrix).

Then, we can easily identify a BFS:

1. Variables associated with I (or permutations of I) are **basic variables**. They take the value of the right hand side vector. Other variables are nonbasic variables; they take values of zero.
2. Columns of the constraint matrix corresponding to the basic variables form a **basis**; we also refer to the set of basic variables sometimes as a basis.

2.5 The Simplex Method

To use the Simplex Method on an LP, we shall assume the following:

1. The LP is in standard form
2. $b \geq 0$
3. There exists a collection B of m variables called a *basis* such that:
 - (a) the submatrix A_B of A consisting of the columns of A corresponding to the variables in B is the $m \times m$ identity matrix
 - (b) the cost coefficients corresponding to the variables in B are all equal to 0.

2.5.1 Phase I

In Phase I of the Simplex Method, we need to transform the LP into a standard form in Equation 10.

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

Appendix & Acknowledgements

These notes are heavily inspired by Stanford's CS229: Machine Learning Notes, as well as Convex Optimisation by Boyd and Vandenberghe.