Contents

1

Linear Programming

Recall that in general, an optimisation problem can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}) \\ &\text{s.t. } g_i(\boldsymbol{x}) = 0 \quad \text{for } i = 1, 2, \cdots, p \\ &\qquad h_j(\boldsymbol{x}) \le 0 \quad \text{for } j = 1, 2, \cdots, m, \end{aligned}$$

where f is known as the *objective function*, g_i's are known as *equality constraints* and h_j's are known as *inequality constraints*. The set of all $\boldsymbol{x}$ that satisfy all constraints is called the *feasible set*, where each of such $\boldsymbol{x}$ is a *feasible solution*. The vector which minimises f is known as the *optimal solution* and is denoted by $\boldsymbol{x}^*$, with $f(\boldsymbol{x}^*)$ being the *optimal value*.

The concept of a *linear program* is intuitive to understand: it is simply an optimisation problem whose objective function and constraint functions are all linear. In a 2-dimensional plane, we see that any region bounded by linear functions is a polygon. We will abstract this idea and generalise it for any finite-dimensional space.

1.1 Geometry of Linear Programming

In any Euclidean space $\mathbb{R}^n$, a linear function can be written as

$$f(x_1, x_2, \cdots, x_n) = a_0 + \sum_{i=1}^{n} a_i x_i,$$

where the a_i's are real coefficients. In matrix notations, this becomes

$$f(\boldsymbol{x}) = \mathbf{c}^{\mathrm{T}}\boldsymbol{x} + a_0$$

for some $\boldsymbol{c} \in \mathbb{R}^n$. Let $f(\boldsymbol{x}) = a_0 + b$, then we have $\mathbf{c}^{\mathrm{T}}\boldsymbol{x} = b$. Note that this level set equation gives a linear function in $\mathbb{R}^{n-1}$, because obviously $\mathbf{c}^{\mathrm{T}}\boldsymbol{x} = c_n x_n + \bar{\mathbf{c}}^{\mathrm{T}}\boldsymbol{x}'$, so

$$x_n = \frac{b}{c_n} - \frac{1}{c_n}\bar{\mathbf{c}}^{\mathrm{T}}\boldsymbol{x}'.$$

Apparently, $-\frac{1}{c_n}\bar{\boldsymbol{c}} \in \mathbb{R}^{n-1}$, so x_n is a linear function of $\boldsymbol{x}'$. This means every straight line in $\mathbb{R}^n$ is defined by an equation

$$\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b$$

for some $\boldsymbol{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Take $\boldsymbol{x}_0 \in \mathbb{R}^n$ which is on this line an let $\boldsymbol{d} \in \mathbb{R}^n$ be the direction vector of the line, then for every $\boldsymbol{x}$ with $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b$, we also have

$$\boldsymbol{x} = \boldsymbol{x}_0 + \lambda\boldsymbol{d}$$

for some $\lambda \in R$. However, this implies that for any such $\boldsymbol{x}$, we have

$$\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{x}_0 + \lambda\boldsymbol{d}) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b,$$

but $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_0 = b$, so we have to have $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{d} = 0$. Therefore, the set

$$A_{\perp} \coloneqq \{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b\}$$

in fact is a set of vectors orthogonal to $\boldsymbol{a}$ in $\mathbb{R}^n$.

Definition 1.1.1 ▸ Hyperplane

Let $\boldsymbol{a} \in \mathbb{R}^n$ be a vector. For any $b \in \mathbb{R}$, the set

$$H_b \coloneqq \{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b\}$$

is said to be a **hyperplane** with normal vector $\boldsymbol{a}$.

It is easy to see that in $\mathbb{R}^2$, a hyperplane is a straight line perpendicular to $\boldsymbol{a}$ and in $\mathbb{R}^3$, it is a plane whose normal vector is parallel to $\boldsymbol{a}$.

Intuitively, a hyperplane partitions the space $\mathbb{R}^n$ into 2 halves. Therefore, we refer to the set

$$\{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b\}$$

as a *half-space*. Intuitively, if we have m half-spaces, then their intersection is a polyhedron in the space, i.e., we define the set

$$P \coloneqq \bigcap_{i=1}^{m} \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i\}$$

as a *polyhedral set*.

Remark. Note that for P to be a polyhedron, the intersection must be finite. Otherwise, consider the counter example of the bounded set whose boundary is defined by all hyperplanes at a distance d away from a fixed point Q, which is a sphere.

We can let $\boldsymbol{a}_i^{\mathrm{T}}$ be the i-th row of the matrix $\boldsymbol{A}$ and define a column vector $\boldsymbol{b}$ whose i-th entry is b_i, then we can re-write the above intersection using matrix multiplication.

Definition 1.1.2 ▶ Polyhedron

A **polyhedron** is defined as the set

$$P := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$.

1.2 Standard Form of Linear Programs

Definition 1.2.1 ▶ Linear Programming Problem

A **linear programming** (LP) problem is an optimisation problem where the objective function f is linear and the feasible set P is a polyhedron.

Note that each linear constraint corresponds to a half-space, so we can formulate a linear programming problem as

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathbb{R}^n} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ &\text{s.t. } \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i \quad \text{for } i = 1, 2, \cdots, p \\ &\qquad \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{x} = b_j \quad \text{for } i = 1, 2, \cdots, m, \\ &\qquad \boldsymbol{a}_k^{\mathrm{T}}\boldsymbol{x} \neq b_k \quad \text{for } i = 1, 2, \cdots, q. \end{aligned}$$

where $\boldsymbol{c} \in \mathbb{R}^n$ is called the *cost* or *profit* vector, $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}$, $\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{x}$ and $\boldsymbol{a}_k^{\mathrm{T}}\boldsymbol{x}$ are called the *constraints* and $\boldsymbol{x}$ is known as *decision variables*. Note that the number of constraints must be finite, or else we may have a feasible set which is not a polyhedron.

Given any linear program, it may present one of the following possibilities:

1. The program has a unique optimal solution.
2. The program has infinitely many optimal solution (but still a unique optimal value).
3. The program has no optimal solution.

4. The program has no feasible solution.

The first 2 cases are trivial. Suppose a linear program with objective function f has no optimal solution, then it means for every feasible $\boldsymbol{x}$, we can find a different feasible $\boldsymbol{x}'$ such that $f(\boldsymbol{x}') < f(\boldsymbol{x})$.

Suppose a linear program with objective function f has no feasible solution, then this means that the feasible set is empty. In this case, we define the optimal value to be ∞. The reasoning is as follows.

Suppose f and g are objective functions of 2 optimisation problems with feasible set S_1 and S_2 respectively such that $S_1 \subseteq S_2$. Clearly, $\min f(\boldsymbol{x}) \geq \min g(\boldsymbol{x})$. Note that if $S_1 = \emptyset$, then for every S_2, we have the above inequality, which means that

$$\min_{\boldsymbol{x} \in S_1} f(\boldsymbol{x}) \geq y$$

for all $y \in \mathbb{R}$. Therefore, $\min_{\boldsymbol{x} \in S_1} f(\boldsymbol{x}) = \infty$.

For each inequality constraint in the form of $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$, we can introduce a *slack variable* $s_i \geq 0$ such that $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} + s_i = b_i$. For each constraint on x_i in the form of $x_i \leq 0$, we can replace every occurrence of x_i by $-x_i^- = x_i$ such that $x_i^- \geq 0$. For each free variable x_j, we can express it as

$$x_j = x_j^+ - x_j^- \quad \text{for some } x_j^+, x_j^- \geq 0.$$

For instance, we can take $x_i^+ = 0$ and $x_i^- > 0$ whenever $x_i < 0$ and vice versa for $x_i > 0$. Note that this correspondence is not unique.

After the above transformations, we see that every constraint is equivalent to either an equality constraint $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}+s_i = b_i$ or an inequality constraint in the form of $x_i \geq 0$. Therefore, we define the following as the *standard form* of a linear program:

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathbb{R}^n} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ &\text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ &\qquad x_i \geq 0, \quad \text{for } i = 1, 2, \cdots, m. \end{aligned}$$

One should realise that a linear program in the standard form can be more easily solved by using linear algebra to find the optimal solution. Note that not every optimisation problem is given in the standard form. Fortunately, we can always convert a linear program into the standard form.

1.3 Convex Sets and Functions

Intuitively, we describe two types of shapes in natural languages: the shapes which, if you choose any of its edges, lies in the same side of that edge, and the shapes which span across both sides from some chosen edge of its.

Graphically, this means that some shapes are "convex" to all directions, where as some other shapes are "concave". We shall define this rigorously as follows:

Definition 1.3.1 ▸ Convex Set

A set $D \subseteq \mathbb{R}^n$ is said to be **convex** if for all $\boldsymbol{x}, \boldsymbol{y} \in D$ and for all $\lambda \in [0, 1]$,

$$\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in D.$$

Analogously, we might want to say that a function is convex if, for any 2 points on its graph, the line segment joining the 2 points "lies within" the graph of the function. We can define convexity over functions as follows:

Definition 1.3.2 ▸ Convex Function

A function $f : D \to \mathbb{R}^n$ is said to be **convex** if for all $\boldsymbol{x}, \boldsymbol{y} \in D$ and for all $\lambda \in [0, 1]$,

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y}).$$

Definition 1.3.3 ▸ Concave Function

A function $f : D \to \mathbb{R}^n$ is said to be **concave** if for all $\boldsymbol{x}, \boldsymbol{y} \in D$ and for all $\lambda \in [0, 1]$,

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \geq \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y}).$$

From another perspective, we can see that for any convex function f, the tangent plane to the graph of f at any point will lie below the graph.

> *Remark.* A function which is not convex must be concave. However, a function which is convex may not be non-concave (consider $f(x) = x$).

Therefore, it is easy to see that all functions in the form of $f(\boldsymbol{x}) = d + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ are both convex and concave. Such functions are said to be *affine* functions. Equivalently, this means that all affine functions are neither strictly convex nor strictly concave.

In the above definitions, the expression $\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}$ for $\lambda \in [0, 1]$ is known as a *convex combination*. This notion can be generalised for any finite number of terms.

Proposition 1.3.4 ▶ Generalised Convex Combination

Let $k \in \mathbb{N}^+$ and let $f : S \to \mathbb{R}$ be a convex function on the convex set $S \subseteq \mathbb{R}^n$ and let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k \in S$, then

$$f\left(\sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i\right) \leq \sum_{i=1}^{k} \lambda_i f(\boldsymbol{x}_i),$$

where $\sum_{i=1}^{k} \lambda_i = 1$ and $\lambda_i \geq 0$ for $i = 1, 2, \cdots, k$.

Using the idea of convex combinations, we can define the notion of a *convex hull*.

Definition 1.3.5 ▶ Convex Hull

The **convex hull** of $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ is defined as the set of all convex combinations of the vectors, denoted by

$$\operatorname{conv}(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) := \left\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i, \lambda_i \in [0, 1], \sum_{i=1}^{n} \lambda_i = 1\right\}.$$

Note that $\operatorname{conv}(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$ is the smallest convex set containing all of $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$.

Now we consider the following proposition:

Proposition 1.3.6 ▶ Maximum of Convex Functions Is Convex

Let $f_1, f_2, \cdots, f_m : \mathbb{R}^n \to \mathbb{R}$ be convex functions, then the function

$$f(\boldsymbol{x}) := \max_{i=1,2,\cdots,m} f_i(\boldsymbol{x})$$

is convex.

Proof. Take any $\boldsymbol{x} \neq \boldsymbol{y} \in \mathbb{R}^n$ and consider $\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}$ for some $\lambda \in [0, 1]$. Note that for each of the f_i's, we have

$$f_i\big(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}\big) \leq \lambda f_i(\boldsymbol{x}) + (1 - \lambda) f_i(\boldsymbol{y}),$$

and so

$$\max_{i=1,2,\cdots,m} f_i\big(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}\big) \leq \max_{i=1,2,\cdots,m} \left[\lambda f_i(\boldsymbol{x}) + (1 - \lambda) f_i(\boldsymbol{y})\right].$$

Therefore,

$$\begin{aligned} f\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) &= \max_{i=1,2,\cdots,m} f_i\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \\ &\le \max_{i=1,2,\cdots,m} [\lambda f_i(\boldsymbol{x}) + (1-\lambda) f_i(\boldsymbol{y})] \\ &= \lambda \max_{i=1,2,\cdots,m} f_i(\boldsymbol{x}) + (1-\lambda) \max_{i=1,2,\cdots,m} f_i(\boldsymbol{y}) \\ &= \lambda f(\boldsymbol{x}) + (1+\lambda) f(\boldsymbol{y}). \end{aligned}$$

□

An immediate corollary of Proposition 1.3.6 allows us to define a piece-wise convex affine function.

Corollary 1.3.7 ▶ Piece-wise Affine Functions Are Convex

The piece-wise affine function

$$f(\boldsymbol{x}) = \max_{i=1,2,\cdots,n} \left(\boldsymbol{c}_i^{\mathrm{T}} \boldsymbol{x} + d_i\right)$$

is convex.

2

The Simplex Method

2.1 Basic Feasible Solutions

Recall that given any linear program, its feasible set is a polyhedron P. Note that in $\mathbb{R}^n$, the smallest possible number of hyperplanes intersecting at a point is n. Intuitively, any polyhedron can be completely defined by all of such "corner points" of itself. Here we provide three equivalent definitions.

Definition 2.1.1 ▶ Extreme Point

Let P be a polyhedron, a point $\boldsymbol{x}^* \in P$ is said to be an **extreme point** if whenever there are $\boldsymbol{y}, \boldsymbol{z} \in P$ with $\boldsymbol{x}^* = \lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{z} = \boldsymbol{x}^*$ for some $\lambda \in (0,1)$, we have $\boldsymbol{y} = \boldsymbol{z} = \boldsymbol{x}^*$.

We can interpret the definition as follows: suppose $\boldsymbol{x}^*$ is not a corner point, then there are 2 possibilities. If $\boldsymbol{x}^*$ is an internal point, then there is some $\delta > 0$ such that the neighbourhood $V_\delta(\boldsymbol{x}^*) \subseteq P$. Therefore, we can always find $\boldsymbol{y}, \boldsymbol{z} \in V_\delta(\boldsymbol{x}^*)$ with $\boldsymbol{y} \neq \boldsymbol{z} \neq \boldsymbol{x}^*$ such that $\boldsymbol{x}^* \in (\boldsymbol{y}, \boldsymbol{z})$. Otherwise, $\boldsymbol{x}^*$ is on the boundary, i.e.,

$$\boldsymbol{x}^* \in H \coloneqq \left\{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b\right\}.$$

Clearly, we can also find $\boldsymbol{y}, \boldsymbol{z} \in V_\delta(\boldsymbol{x}^*) \cap H$ with $\boldsymbol{y} \neq \boldsymbol{z} \neq \boldsymbol{x}^*$ such that $\boldsymbol{x}^* \in (\boldsymbol{y}, \boldsymbol{z})$.

Alternatively, we consider a hyperplane defined by $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} = b$ such that $\boldsymbol{c}$ is not orthogonal to any of the boundaries of the polyhedron P. Clearly, $\boldsymbol{c}$ is the gradient vector of the affine function

$$f(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}.$$

Therefore, by translating the hyperplane in the direction of $\boldsymbol{c}$, we are able to maximise b. Note that this hyperplane is not parallel to any boundary of the polyhedron, so when b reaches the maximum, the hyperplane will have a unique intersection with the polyhedron, which must be a corner point.

Definition 2.1.2 ▶ Vertex

Let P be a polyhedron, a point $\boldsymbol{x}^* \in P$ is said to be a **vertex** if there exists some $\boldsymbol{c}$ such that $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^* > \boldsymbol{c}^T\boldsymbol{y}$ for all $\boldsymbol{y} \in P - \{\boldsymbol{x}^*\}$.

Note that here we require the inequality to be strict, because otherwise $\boldsymbol{x}^*$ and $\boldsymbol{y}$ may both be internal points of some boundary hyperplane of P.

Note that any boundary of a polyhedron P is uniquely determined by a constraint $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$. Let $\boldsymbol{x}^* \in P$, it is clear that $\boldsymbol{x}^*$ is "on the boundary" if and only if $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$ for some i. In such cases, we say that the corresponding constraint is *active/binding/tight* at $\boldsymbol{x}^*$.

Geometrically, a corner point of a polyhedron P in $\mathbb{R}^n$ is the intersection of n boundary hyperplanes which are pair-wise non-parallel. Let $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$ and $\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{x} = b_j$ be any of the n hyperplanes, then it is clear that $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ must be linearly independent for the two hyperplanes to be non-parallel. This leads to the notion of *basic feasible solutions*.

Definition 2.1.3 ▸ Basic Feasible Solution

Let $P \subseteq \mathbb{R}^n$ be a polyhedron. $\boldsymbol{x}^* \in P$ is said to be a **basic feasible solution** if there are n linearly independent constraints which are active at $\boldsymbol{x}^*$.

We can generalise Definition 2.1.3 to deal with even infeasible points. First we introduce some terminologies.

Definition 2.1.4 ▸ Rank

Let $P \subseteq \mathbb{R}^n$ be a polyhedron with constraints $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$ for $i = 1, 2, \cdots, m$. For any $\boldsymbol{x} \in \mathbb{R}^n$, the **rank** of $\boldsymbol{x}$ is defined as

$$\operatorname{rank}(\boldsymbol{x}) \coloneqq \dim\left(\operatorname{span}\left\{\boldsymbol{a}_j : \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{x} = b_j\right\}\right).$$

Clearly, $\boldsymbol{x}^* \in \mathbb{R}^n$ is a basic feasible solution if and only if $\operatorname{rank}(\boldsymbol{x}^*) = n$ and $\boldsymbol{x}^* \in P$. Notice that not all $\boldsymbol{x}$ with rank n is feasible, so we might consider the following definition:

Definition 2.1.5 ▸ Basic Solution

Let $P \subseteq \mathbb{R}^n$ be a polyhedron. A vector $\boldsymbol{x} \in \mathbb{R}^n$ is a **basic solution** if $\operatorname{rank}(\boldsymbol{x}) = n$.

One important thing to note here is that $\operatorname{rank}(\boldsymbol{x}) = n$ does not necessarily imply that there are exactly n constraints active at $\boldsymbol{x}$. Intuitively, there can be more than n hyperplanes in $\mathbb{R}^n$ which intersect at a point $\boldsymbol{x}$. However, if $\operatorname{rank}(\boldsymbol{x}) = n$, then some hyperplanes are "redundant", i.e., their normal vectors can be expressed as linear combinations of the gradients of some n linearly independent constraints.

Definition 2.1.6 ▸ Degeneracy

A basic solution $\boldsymbol{x} \in \mathbb{R}^n$ is said to be **degenerate** if there are more than n constraints active at $\boldsymbol{x}$.

Let $P \subseteq \mathbb{R}^n$ be a polyhedron defined by m constraints. Clearly, for each basic feasible solution, we require at least n different constraints to be active. Therefore, the number of distinct basic feasible solutions in P is at most $\binom{m}{n}$. This justifies the fact that any polyhedron in $\mathbb{R}^n$ determined by less than n constraints has no basic feasible solution, and any polyhedron in a finite-dimensional space must have finitely many basic feasible solutions.

Suppose we are given a standard linear program, then we can write its feasible set as the polyhedron

$$P := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. Obviously, P has no basic feasible solution if $m < n$. Suppose $m \geq n$, we will devise a way to compute the basic feasible solutions systematically.

Theorem 2.1.7 ▸ Basic Solution Characterisation

Let

$$P := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$$

be a polyhedron for some $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$. A vector $\boldsymbol{x}^* \in \mathbb{R}^n$ is a basic solution if and only if

- $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$, *and*
- *There exists an index set $B \subseteq \{1, 2, \cdots, n\}$ such that the set*

  $$\{\boldsymbol{A}_i : i \in B\}$$

  is linearly independent and $x_j^* = 0$ for all $j \notin B$, where

  $$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \cdots & \boldsymbol{A}_n \end{bmatrix}.$$

Proof. Write $B = \{B(1), B(2), \cdots, B(m)\}$ and define

$$N = \{N(1), N(2), \cdots, N(n-m)\} := \{1, 2, \cdots, n\} - B.$$

For each $i \in N$, since $x_i^* = 0$, we have $\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{x}^* = 0$. Therefore, the matrix representation for the active constraints is

$$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{e}_{N(1)}^{\mathrm{T}} \\ \boldsymbol{e}_{N(2)}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_{N(n-m)}^{\mathrm{T}} \end{bmatrix} \boldsymbol{x}^* = \boldsymbol{0}.$$

Re-arranging the columns, the above matrix can be re-written as

$$\begin{bmatrix} \boldsymbol{A}_B & \boldsymbol{A}_N \\ \boldsymbol{0} & \boldsymbol{I}_N \end{bmatrix} \bar{\boldsymbol{x}}^* = \boldsymbol{0},$$

where $\bar{\boldsymbol{x}}^*$ is obtained by re-arranging the rows of $\boldsymbol{x}^*$ accordingly. Note that the columns of $\boldsymbol{A}_B$ is linearly independent, so $\det(\boldsymbol{A}_B) \neq 0$. Therefore,

$$\begin{vmatrix} \boldsymbol{A}_B & \boldsymbol{A}_N \\ \boldsymbol{0} & \boldsymbol{I}_N \end{vmatrix} = \det(\boldsymbol{A}_B)\det(\boldsymbol{I}_N) \neq 0,$$

and so the matrix is invertible. Therefore, the rows of the matrix are linearly independent. This means that there are n linearly independent constraints active at $\boldsymbol{x}^*$. Therefore, $\boldsymbol{x}^*$ is a basic feasible solution.

Suppose conversely that $\boldsymbol{x}^*$ is a basic feasible solution, then clearly $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$. Since there are m equality constraints, then we must have $(n-m)$ active active inequality constraints at $\boldsymbol{x}^*$, indexed by $N = \{N(1), N(2), \cdots, N(n-m)\}$, such that the constraints are linearly independent. Therefore, the matrix

$$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{e}_{N(1)}^{\mathrm{T}} \\ \boldsymbol{e}_{N(2)}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_{N(n-m)}^{\mathrm{T}} \end{bmatrix}$$

is invertible and that for all $i \in N$, $x_i^* = 0$. Let

$$B = \{B(1), B(2), \cdots, B(m)\} := \{1, 2, \cdots, n\} - N$$

be an index set, then the above matrix can be re-arranged as

$$\begin{bmatrix} \boldsymbol{A}_B & \boldsymbol{A}_N \\ \boldsymbol{0} & \boldsymbol{I}_N \end{bmatrix},$$

which is invertible. Therefore, $\left\{\boldsymbol{A}_{B(1)}, \boldsymbol{A}_{B(2)}, \cdots, \boldsymbol{A}_{B(m)}\right\}$ is linearly independent. □

Note that according to Theorem 2.1.7, for every $i \in N$, $x_i = 0$. Note that $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_B & \boldsymbol{A}_N \end{bmatrix}$, so the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ is equivalent to $\boldsymbol{A}_B\boldsymbol{x}_B = \boldsymbol{b}$. Therefore, given any standard polyhedron $P := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$, we can construct a basic solution in P using the following procedures:

1. Choose m linearly independent columns from $\boldsymbol{A}$ with B being the index set for the columns to form the set $\{\boldsymbol{A}_{B(1)}, \boldsymbol{A}_{B(2)}, \cdots, \boldsymbol{A}_{B(m)}\}$;
2. For each $i \in N := \{1, 2, \cdots, n\} - B$, set $x_i = 0$. The vector consisting of all of these zero entries is denoted by $\boldsymbol{x}_N$;
3. Solve the linear system $\boldsymbol{A}_B\boldsymbol{x}_B = \boldsymbol{b}$ to obtain $\boldsymbol{x}_B = \boldsymbol{A}_B^{-1}\boldsymbol{b}$;
4. Let x_i be the i-th entry of $\boldsymbol{x}$, then

$$x_i = \begin{cases} 0, & \text{if } i \in N \\ (\boldsymbol{x}_B)_i & \text{if } i \in B \end{cases}.$$

The $\boldsymbol{x}$ obtained this way is alternatively denoted as $\boldsymbol{x} := (\boldsymbol{x}_B, \boldsymbol{x}_N)$.

Geometrically, 2 distinct basic feasible solutions of P are *adjacent* if there is an edge on the boundary joining the 2 points. Here we give an equivalent definition algebraically.

Definition 2.1.8 ▸ Adjacency

Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be distinct basic solutions with respect to polyhedron P. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are said to be **adjacent** if there are exactly $(n-1)$ linearly independent constraints active at both points, or their corresponding bases only contain 1 different basic column.

Recall that not all polyhedrons have basic feasible solutions. Informally, we can see that if a polyhedron does not contain any basic feasible solution, it contains at least 2 "openings" which allows us to place a straight line into the polyhedron. This observation is summarised rigorously as follows:

Theorem 2.1.9 ▸ Conditions for the Existence of Basic Feasible Solution

Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *and*

$$P := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{Ax} \leq \boldsymbol{b}\} \neq \varnothing,$$

then the following statements are equivalent:

1. *P does not contain any straight line.*
2. *P has a basic feasible solution.*
3. *P has n linearly independent constraints.*

Proof. Suppose P has a basic feasible solution $\boldsymbol{x}^*$, then there are n linearly independent constraints active at $\boldsymbol{x}^*$. Therefore, it is trivial that P must contain at least n linearly independent constraints.

We shall prove that (3) implies (1) by considering the contrapositive statement. Suppose that P contains a straight line $\{\boldsymbol{x}_0 + \lambda \boldsymbol{d} : \lambda \in \mathbb{R}\}$ for some fixed point $\boldsymbol{x}_0 \in P$ and direction vector $\boldsymbol{d} \in \mathbb{R}^n$, then for any $\lambda \in \mathbb{R}$, we have $\boldsymbol{A}(\boldsymbol{x}_0 + \lambda \boldsymbol{d}) \leq \boldsymbol{b}$. Notice that $\boldsymbol{A}\boldsymbol{x}_0 \leq \boldsymbol{b}$, so $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$. However, $\boldsymbol{d} \neq \boldsymbol{0}$, so $\boldsymbol{A}$ does not contain n independent rows, which implies that P does not have n linearly independent constraints.

Suppose that P does not contain any straight line. Take some $\boldsymbol{x} \in P$ and let

$$I(\boldsymbol{x}) := \left\{\boldsymbol{a}_i : \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x} = b_i\right\}$$

be the set of gradient vectors of all active constraints at $\boldsymbol{x}$. If $I(\boldsymbol{x})$ contains n linearly independent vectors, then we are done. Otherwise, $I(\boldsymbol{x})$ does not span $\mathbb{R}^n$, so there is some $\boldsymbol{d} \in \mathbb{R}^n$ with $\boldsymbol{d} \neq 0$ such that it is normal to $\operatorname{span}(I(\boldsymbol{x}))$. Since P contains no straight lines, there exists some constraints with gradient vector $\boldsymbol{a}_j \notin I(\boldsymbol{x})$ such that we can find some $\mu \in \mathbb{R}$ such that $\boldsymbol{a}_j^{\mathrm{T}}(\boldsymbol{x} + \mu \boldsymbol{d}) = b_j$. Set $\boldsymbol{x}' = \boldsymbol{x} + \mu \boldsymbol{d}$. Note that for all $\boldsymbol{a}_i \in I(\boldsymbol{x})$, we have $\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{d} = 0$, and so $\boldsymbol{a}_i^{\mathrm{T}}(\boldsymbol{x} + \mu \boldsymbol{d}) = b_i$. Therefore, $\boldsymbol{a}_i \in I(\boldsymbol{x}')$, and so we have $I(\boldsymbol{x}) \cup \{\boldsymbol{a}_j\} \subseteq I(\boldsymbol{x}')$. We claim that $\boldsymbol{a}_j$ linearly independent with $I(\boldsymbol{x})$. Otherwise, $\boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{d} = 0$, which means that the boundary defined by $\boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} = b_j$ is parallel to $\boldsymbol{d}$ and so if the constraint is active at $\boldsymbol{x}'$, it must also be active at $\boldsymbol{x}$, which is a contradiction. Therefore, $I(\boldsymbol{x}')$ has more linearly independent vectors than $I(\boldsymbol{x})$. Repeat this process and we will eventually obtain a set of n linearly independent gradient vectors, $I(\boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ is a basic feasible solution. □

Consider the level set defined by $f(\boldsymbol{x}) = k$ for some affine function f. Suppose the level set has an intersection with some polyhedron P, then by translating the level set in the direction of $-\nabla f$, it will eventually intersect P at a basic feasible solution such that any further translation will make the intersection empty. Therefore, a reasonable guess is that an optimal solution for any linear program is closely linked to the basic feasible solutions of the feasible set.

Theorem 2.1.10 ▸ Optimality of Basic Feasible Solutions

Consider the linear program

$$\min_{x \in P} f(\boldsymbol{x})$$

where $f(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ and P is a polyhedron. If P has a basic feasible solution and the linear program has an optimal solution, then there exists a basic feasible solution of P which is an optimal solution to the linear program.

Proof. Let the optimal value of f be v^*, then the set of optimal solutions is

$$Q := P \cap \left\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} = v^*\right\}.$$

Note that Q is a polyhedron. By Theorem 2.1.9, since P contains a basic feasible solution, it does not contain a straight line. Therefore, $Q \subseteq P$ cannot contain a straight line and so it contains a basic feasible solution.

Let $\boldsymbol{x}^*$ be any basic feasible solution of Q. We claim that $\boldsymbol{x}^*$ is a basic feasible solution of P. Suppose on contrary that $\boldsymbol{x}^*$ is not a basic feasible solution of P, then there exists some $\boldsymbol{d} \in \mathbb{R}^n$ with $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{d} = 0$ such that $\boldsymbol{x}^* + \lambda\boldsymbol{d} \in Q$ is an internal point of P for some $\lambda \in \mathbb{R}$. However, this means that there exists some $\mu > 0$ such that $\boldsymbol{x}^* - \boldsymbol{c} \in P$. Note that $\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{c}) < v*$, which is impossible. Therefore, $\boldsymbol{x}^*$ must be a basic feasible solution of P. □

Note that this essentially implies that Q is the convex hull of basic feasible solutions of P. Therefore, any optimal solution of the original linear program can be expressed as a convex combination of basic feasible solutions of P, and so it suffices to first find all basic feasible solutions if our feasible set contains any.

2.2 The Simplex Method

Note that for any linear program, we can convert it to a standard linear program with feasible region

$$P := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}.$$

Let $\{\boldsymbol{x}_0 + \lambda\boldsymbol{d} : \lambda \in \mathbb{R}\}$ be any straight line, then clearly we can find some $\lambda \in \mathbb{R}$ such that there is some entry x_i of $\boldsymbol{x}$ with $x_i < 0$. Therefore, P does not contain any straight line, and so P always contains at least one basic feasible solution if $P \neq \varnothing$. Therefore, as long as a linear program has a finite optimal value, it has optimal solutions which are basic feasible solutions.

Therefore, to find the optimal solution of a linear program, we only need to start at any basic feasible solution and try to reach an adjacent basic feasible solution which can improve our objective value. Continue this search and eventually we will be able to collect all optimal solutions.

Given any basic feasible solution $\boldsymbol{x}$, we wish to find a direction $\boldsymbol{d}$ such that $\boldsymbol{x} + \theta\boldsymbol{d}$ is some adjacent basic feasible solution for some $\theta \in \mathbb{R}$. First, we need to ensure that $\boldsymbol{x} + \theta\boldsymbol{d}$ is still feasible.

Definition 2.2.1 ▸ Feasible Direction

Let P be a polyhedron and $\boldsymbol{x} \in P$ be a feasible point. A vector $\boldsymbol{d}$ is a **feasible direction** if $\boldsymbol{x} + \lambda\boldsymbol{d} \in P$ for some $\lambda > 0$.

Notice that $\boldsymbol{x} + \lambda\boldsymbol{d}$ is still feasible, so we must have $\boldsymbol{A}(\boldsymbol{x} + \lambda\boldsymbol{d}) = \boldsymbol{b}$. However, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, which implies that $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$. We can write $\boldsymbol{d} = (\boldsymbol{d}_B, \boldsymbol{d}_N)$ with respect to $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_N)$. We need $\boldsymbol{x} + \lambda\boldsymbol{d} \geq \boldsymbol{0}$, but $\boldsymbol{x}_N = \boldsymbol{0}$, so we must also have $\boldsymbol{d}_N \geq \boldsymbol{0}$.

Suppose $\boldsymbol{d}$ is a feasible direction and $\boldsymbol{x}$ is a basic feasible solution. Consider $\boldsymbol{x}' = \boldsymbol{x} + \theta\boldsymbol{d} \in P$ for some $\theta \in \mathbb{R}$. If $\boldsymbol{x}'$ is on an edge, then clearly there is exactly one less active constraint at $\boldsymbol{x}'$ than at $\boldsymbol{x}$.

Theorem 2.2.2 ▸ Characterisation of A Direction Connecting Basic Feasible Solutions

Let $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_N)$ with $\boldsymbol{x}_B \geq \boldsymbol{0}$ and $\boldsymbol{x}_N = \boldsymbol{0}$ be a basic feasible solution, then a direction that connects $\boldsymbol{x}$ to an adjacent basic feasible solution is in the form of $\boldsymbol{d}^j = \left(\boldsymbol{d}_B^j, \boldsymbol{d}_N^j\right)$ for some $j \in N$, such that $\boldsymbol{d}_N^j = \boldsymbol{e}_j$ and $\boldsymbol{d}_B^j = -\boldsymbol{A}_B^{-1}\boldsymbol{A}_j$.

Proof. Note that for any feasible $\boldsymbol{x}' = \boldsymbol{x} + \theta\boldsymbol{d}^j$ which is not a basic feasible solution, we have $\boldsymbol{A}\boldsymbol{x}'_B = \boldsymbol{b}$. This means that all constraints corresponding to B are active along the edge. Therefore, the edge frees one constraint of the form $x_j \geq 0$ for some $j \in N$. This means that the index set of active constraints along the edge is given by $B \cup N - \{j\}$. Note that

$$\boldsymbol{x} + \theta\boldsymbol{d}^j = \left(\boldsymbol{x}_B + \theta\boldsymbol{d}_B^j, \boldsymbol{x}_N + \theta\boldsymbol{d}_N^j\right).$$

Notice that $\boldsymbol{x}_N = \boldsymbol{0}$ and $\boldsymbol{x}_N + \theta\boldsymbol{d}_N^j = \theta\boldsymbol{e}_j$, so $\boldsymbol{d}_N^j = \boldsymbol{e}_j$. Since $\boldsymbol{x} + \theta\boldsymbol{d}^j$ is feasible, we have

$$\boldsymbol{A}\left(\boldsymbol{x} + \theta\boldsymbol{d}^j\right) = \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}.$$

Therefore, $\boldsymbol{A}\boldsymbol{d}^j = \boldsymbol{0}$. However, note that

$$\begin{aligned} \boldsymbol{A}\boldsymbol{d}^j &= \begin{bmatrix} \boldsymbol{A}_B & \boldsymbol{A}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_B^j \\ \boldsymbol{d}_N^j \end{bmatrix} \\ &= \boldsymbol{A}_B\boldsymbol{d}_B^j + \boldsymbol{A}_N\boldsymbol{e}_j \\ &= \boldsymbol{A}_B\boldsymbol{d}_B^j + \boldsymbol{A}_j. \end{aligned}$$

Since $\boldsymbol{A}_B$ has linearly independent columns, it is invertible, so $\boldsymbol{d}_B^j = -\boldsymbol{A}_B^{-1}\boldsymbol{A}_j$. □

Recall that the feasible polyhedron is actually the convex hull of all of its basic feasible solutions. Naturally, we may conjure that any feasible direction is a linear combination of all the $\boldsymbol{d}^j$'s for $j \in N$.

Proposition 2.2.3 ▸ Feasible Directions as Linear Combinations

Let $\boldsymbol{x} := (\boldsymbol{x}_B, \boldsymbol{x}_N)$ *be a basic feasible solution, then any feasible direction at* $\boldsymbol{x}$ *can be expressed as*

$$\boldsymbol{d} = \sum_{j \in N} \lambda_j \boldsymbol{d}^j$$

for $\lambda_j \geq 0$.

Proof. Note that since $\boldsymbol{d}$ is a feasible direction, $\boldsymbol{A}\boldsymbol{d} = \mathbf{0}$. Since $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{A}_B\boldsymbol{d}_B + \boldsymbol{A}_N\boldsymbol{d}_N$, this implies that

$$\boldsymbol{A}_B\boldsymbol{d}_B = -\boldsymbol{A}_N\boldsymbol{d}_N = -\sum_{j \in N} d_j \boldsymbol{A}_j.$$

Note that $\boldsymbol{A}_B$ is invertible, so by Theorem 2.2.2,

$$\boldsymbol{d}_B = -\sum_{j \in N} d_j \boldsymbol{A}_B^{-1}\boldsymbol{A}_j = \sum_{j \in N} d_j \boldsymbol{d}_B^j.$$

Since $\boldsymbol{d}_N^j = \boldsymbol{e}_j$, we have $\boldsymbol{d}_N = \sum_{j \in N} d_j \boldsymbol{d}_N^j$. Take $\lambda_j = d_j$, we have

$$\boldsymbol{d} = \sum_{j \in N} \lambda_j \boldsymbol{d}^j.$$

□

Simply traversing between basic feasible solutions is not very useful, because our ultimate goal is to minimise our objective function, i.e., we wish to find a direction $\boldsymbol{d}^j$ such that for some $\theta > 0$,

$$\boldsymbol{c}^{\mathrm{T}}\left(\boldsymbol{x} + \theta\boldsymbol{d}^j\right) < \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}.$$

Clearly, we require $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{d}^j < 0$. Note that

$$\begin{aligned} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{d}^j &= (\boldsymbol{c}_B, \boldsymbol{c}_N)^{\mathrm{T}} \begin{bmatrix} \boldsymbol{d}_B^j \\ \boldsymbol{d}_N^j \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{c}_B^{\mathrm{T}} & \boldsymbol{c}_N^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} -\boldsymbol{A}_B^{-1}\boldsymbol{A}_j \\ \boldsymbol{e}_j \end{bmatrix} \\ &= -\boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_j + c_j, \end{aligned}$$

so our target is simply $c_j - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_j < 0$.

Definition 2.2.4 ▶ Reduced Cost

Let $\boldsymbol{x}$ be a basic feasible solution with respect to objective function $f(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$. Let $\boldsymbol{c} = (\boldsymbol{c}_B, \boldsymbol{c}_N)$. For each $j = 1, 2, \cdots, n$, the **reduced cost** of variable x_j is defined as

$$\bar{c}_j = c_j - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_j.$$

Remark. For any $i \in B$, note that $\boldsymbol{A}_i = \boldsymbol{A}_B\boldsymbol{e}_i$, we have $\boldsymbol{A}_B^{-1}\boldsymbol{A}_i = \boldsymbol{e}_i$. Therefore, $\bar{c}_i = 0$.

We say that a direction $\boldsymbol{d}^j$ is an *improving direction* if and only if $\bar{c}_j < 0$. Furthermore, notice that by Theorem 2.2.2, we have

$$\begin{aligned}
\bar{c}_j &= c_j - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_j \\
&= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{e}_j - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_j \\
&= \boldsymbol{c}_N^{\mathrm{T}}\boldsymbol{d}_N^j - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{d}_B^j \\
&= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{d}^j.
\end{aligned}$$

Combining with Proposition 2.2.3, the above gives us a quicker way to compute the reduced cost given any feasible direction at a point $\boldsymbol{x}$.

Next, we only need to determine a $\bar{\theta}_j > 0$ such that $\boldsymbol{x} + \bar{\theta}_j\boldsymbol{d}^j$ gives us another basic feasible solution. Since we already know that $\boldsymbol{A}\left(\boldsymbol{x} + \bar{\theta}_j\boldsymbol{d}^j\right) = \boldsymbol{b}$ for all $\bar{\theta}_j > 0$, a natural idea is to take $\bar{\theta}_j$ to be the greatest positive real number such that $\boldsymbol{x} + \bar{\theta}_j\boldsymbol{d}^j \geq \boldsymbol{0}$, i.e., we will move in the direction $\boldsymbol{d}^j$ until we can barely stay in the feasible polyhedron. Consider

$$\boldsymbol{x} + \bar{\theta}_j\boldsymbol{d}^j = \left(\boldsymbol{x}_B + \bar{\theta}_j\boldsymbol{d}_B^j, \boldsymbol{x}_N + \bar{\theta}_j\boldsymbol{d}_N^j\right).$$

Note that $\boldsymbol{x}_N \geq \boldsymbol{0}$ and $\bar{\theta}_j\boldsymbol{d}_N^j = \bar{\theta}_j\boldsymbol{e}_j \geq \boldsymbol{0}$, so it suffices to check that $\boldsymbol{x}_B + \bar{\theta}_j\boldsymbol{d}_B^j \geq \boldsymbol{0}$. This implies that for each $i \in B$, we have $\bar{\theta}_j \geq -\frac{x_i}{d_i^j}$. Therefore, we only need to take

$$\bar{\theta}_j = \min\left\{-\frac{x_i}{d_i^j} : i \in B, d_i^j < 0\right\}.$$

Note that here we do not consider those d_i^j's with $d_i^j > 0$ because in such cases $x_i + \bar{\theta}_j d_i^j \geq 0$ for all $\bar{\theta}_j > 0$. Now, we would verify that the new point we reach is indeed another basic feasible solution.

Proposition 2.2.5 ▸ $\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j$ Is A Basic Feasible Solution

If $\left\{i \in B : d_i^j < 0\right\} \neq \varnothing$, then $\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j$ is a basic feasible solution adjacent to $\boldsymbol{x}$.

Proof. Note that $\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j$ is feasible, so $\boldsymbol{A}\left(\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j\right) = \boldsymbol{b}$. Note that by the definition of $\bar{\theta}_j$, we have

$$\bar{\theta}_j = -\frac{x_\ell}{d_\ell^j}$$

for some $\ell \in B$. Therefore, we have $\left(\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j\right)_\ell = 0$. Notice that at $\boldsymbol{x} \coloneqq (\boldsymbol{x}_B, \boldsymbol{x}_N)$, we have $\ell \in B$ and $j \in N$. Consider $\bar{B} \coloneqq (B - \{\ell\}) \cup \{j\}$ and $\bar{N} \coloneqq (N - \{j\}) \cup \{\ell\}$. One may check that $\bar{B}$ is linearly independent, so by Theorem 2.1.7, $\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j$ is a basic feasible solution. □

By now, we already have devised a systematic method to reach another basic feasible solution from a given basic feasible solution such that our objective value improves. The next goal is to determine a condition by which we should terminate our search and declare the current basic feasible solution to be optimal.

Before we do that, we need to first deal with an edge case where some basic feasible solution we find might be degenerate. Note that in the standard form, we have exactly m equality constraints represented by $\boldsymbol{A}_B \boldsymbol{x}_B = \boldsymbol{b}$ which are always active, and at least $(n - m)$ active inequality constraints in the form of $\boldsymbol{x}_N = \boldsymbol{0}$. However, note that $\boldsymbol{x} \geq \boldsymbol{0}$ represents n inequality constraints, so in total we actually have $(m + n)$ constraints. An implication here is that if any entry of $\boldsymbol{x}_B$ is 0, then we will have more than n active constraints at $\boldsymbol{x}$.

Definition 2.2.6 ▸ Degeneracy in Standard Form

Let $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_N)$ be a basic feasible solution. We say that $\boldsymbol{x}$ is **degenerate** if there is some entry of $\boldsymbol{x}_B$ being 0.

Remark. This also concludes that $\boldsymbol{x}$ is non-degenerate if $\boldsymbol{x}_B = \boldsymbol{A}_B^{-1} \boldsymbol{b} > \boldsymbol{0}$.

Recall that along an improving direction $\boldsymbol{d}^j$, the reduced cost for x_j is $\bar{c}_j < 0$. A natural observation here is that if we cannot find any improving direction at a point $\boldsymbol{x}^*$, then this point must be optimal. Therefore, we are tempted to conclude that $\bar{c}_j \geq 0$ for all $j \in N$.

Theorem 2.2.7 ▸ Optimality Conditions for Simplex Method

Let $\boldsymbol{x}^* = (\boldsymbol{x}_B, \boldsymbol{x}_N)$ be a basic feasible solution to some standard linear program. Let $\bar{\boldsymbol{c}}$ be the vector of reduced costs associated with $\boldsymbol{x}^*$, then

1. *If $\bar{\boldsymbol{c}} \geq \boldsymbol{0}$, then $\boldsymbol{x}^*$ is optimal;*

2. *If $\boldsymbol{x}^*$ is optimal and non-degenerate, then $\bar{\boldsymbol{c}} \geq \mathbf{0}$.*

Proof. Suppose that $\bar{\boldsymbol{c}} \geq \mathbf{0}$. Let $\boldsymbol{y}$ be any feasible solution, then $\boldsymbol{y} - \boldsymbol{x}^*$ is a feasible direction. By Proposition 2.2.3,

$$\boldsymbol{y} - \boldsymbol{x}^* = \sum_{j \in N} \lambda_j \boldsymbol{d}^j,$$

where $\lambda_j \geq 0$ for all $j \in N$. Therefore,

$$\begin{aligned} \boldsymbol{c}^{\mathrm{T}} \boldsymbol{y} &= \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^* + \sum_{j \in N} \lambda_j \boldsymbol{c}^{\mathrm{T}} \boldsymbol{d}^j \\ &= \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^* + \sum_{j \in N} \lambda_j \bar{c}_j \\ &\geq \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^*. \end{aligned}$$

Therefore, $\boldsymbol{x}^*$ is an optimal solution.

Suppose that $\boldsymbol{x}^*$ is a non-degenerate optimal solution. For each $j \in N$, consider the adjacent basic feasible solution $\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j$. Notice that

$$\boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}} \left(\boldsymbol{x} + \bar{\theta}_j \boldsymbol{d}^j \right) = \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} + \bar{\theta}_j \bar{c}_j$$

for all $j \in N$. Therefore, we must have $\bar{\theta}_j \bar{c}_j \geq 0$ for all $j \in N$. Since $\boldsymbol{x}$ is non-degenerate, for each $j \in N$ we have

$$\bar{\theta}_j = -\frac{x_\ell}{d_\ell^j} > 0,$$

and so $\bar{c}_j \geq 0$. Note that for all $i \in B$, we have $\bar{c}_i = 0$, so $\bar{\boldsymbol{c}} \geq \mathbf{0}$. □

Recall that $\bar{\boldsymbol{c}}_B = \mathbf{0}$, so it suffices to check that $\bar{\boldsymbol{c}}_N \geq \mathbf{0}$ for the $(n - m)$ inequality constraints to determine whether $\boldsymbol{x}^*$ is an optimal solution.

With the above preliminary works done, we are now able to develop the algorithm for simplex method.

Technique 2.2.8 ▶ Simplex Method

Let $f : \mathbb{R}^n \to \mathbb{R}$ defined by $f(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}$ be the objective function of a standard linear program with feasible set

$$P \coloneqq \{ \boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \mathbf{0} \}.$$

The simplex method finds the optimal solution using the following procedures:

1. Initialise $\boldsymbol{x}_0$ to be any basic feasible solution.
2. At the k-th iteration, choose an index set B_k such that the columns of $\boldsymbol{A}_{B_k}$ are a basis for the column space of $\boldsymbol{A}$.
3. Let $N_k := \{1, 2, \cdots, n\} - B_k$. For each $j \in N_k$, compute the reduced cost

$$\bar{c}_j = c_j - \boldsymbol{c}_{B_k}^{\mathrm{T}} \boldsymbol{A}_{B_k}^{-1} \boldsymbol{A}_j.$$

4. If $\bar{c}_j \geq 0$ for all $j \in N_k$:
   - $\boldsymbol{x}_k$ is an optimal solution.
5. Otherwise:
   (a) Take some $j \in N_k$ such that $\bar{c}_j < 0$. $(\boldsymbol{x}_k)_j$ is called an **entering variable**.
   (b) Compute $\boldsymbol{d}_{B_k}^j = -\boldsymbol{A}_{B_k}^{-1} \boldsymbol{A}_j$.
   (c) If $\boldsymbol{d}_{B_k}^j \geq \boldsymbol{0}$:
   - the problem is **unbounded**.

   (d) Otherwise:
   i. Let $\ell \in B_k$ be such that $\bar{\theta}_j = \min\left\{-\frac{x_i}{d_i^j} : i \in B, d_i^j < 0\right\} = \frac{x_\ell}{d_\ell^j}$. $(\boldsymbol{x}_k)_\ell$ is called a **leaving variable**.
   ii. Update $B_{k+1} := (B - \{\ell\}) \cup \{j\}$.
   iii. Update $\boldsymbol{x}_{k+1}$ by

$$(\boldsymbol{x}_{k+1})_i = \begin{cases} (\boldsymbol{x}_k)_i + \bar{\theta}_j \boldsymbol{d}_i^j & \text{if } i \in B - \{\ell\} \\ \bar{\theta}_j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

2.3 Tableau Implementation

A common way to run simplex method is by tabulating the relevant variables, solutions and reduced costs so that we can keep track of their values conveniently. A generalised tableau looks like the following:

Basic	$\boldsymbol{x}$	Solution
$\bar{\boldsymbol{c}}$	$\boldsymbol{c}^T - \boldsymbol{c}_B^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}$	$-\boldsymbol{c}_B^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{b}$
$\boldsymbol{x}_B$	$\boldsymbol{A}_B^{-1} \boldsymbol{A}$	$\boldsymbol{A}_B^{-1} \boldsymbol{b}$

If we expand the tableau with more detailed information, we get the following representation:

Basic	x_1	$\cdots$	x_n	Solution
$\bar{\boldsymbol{c}}$	$\bar{c}_1$	$\cdots$	$\bar{c}_n$	$-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}_B$
$x_{B(1)}$ $\vdots$ $x_{B(i)}$ $\vdots$ $x_{B(m)}$	$\boldsymbol{A}_B^{-1}\boldsymbol{A}_1$	$\cdots$	$\boldsymbol{A}_B^{-1}\boldsymbol{A}_n$	$\boldsymbol{A}_B^{-1}\boldsymbol{b}$

Notice that $\boldsymbol{A}_B^{-1}\boldsymbol{A}_B = \boldsymbol{I}$, we may wish to re-arrange the columns of the tableau into the following form:

Basic	$\boldsymbol{x}_B$	$\boldsymbol{x}_N$	Solution
$\bar{\boldsymbol{c}}$	$\boldsymbol{0}$	$\boldsymbol{c}_N^T - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_N$	$-\boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{b}$
$\boldsymbol{x}_B$	$\boldsymbol{I}$	$\boldsymbol{A}_B^{-1}\boldsymbol{A}_N$	$\boldsymbol{A}_B^{-1}\boldsymbol{b}$

Note that at each iteration, we need to "swap" the ℓ-th and j-th columns to update B. Essentially, suppose $\ell = B(i)$, then this is equivalent to performing row operations on the matrix

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{c}_N^T - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A}_N & -\boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{b} \\ \boldsymbol{I} & \boldsymbol{A}_B^{-1}\boldsymbol{A}_N & \boldsymbol{A}_B^{-1}\boldsymbol{b} \end{bmatrix}$$

such that the j-th column becomes $\boldsymbol{e}_i$. Then, we will swap the row label $\boldsymbol{x}_\ell$ with the column label $\boldsymbol{x}_j$. We repeat this process until $\bar{\boldsymbol{c}} \geq \boldsymbol{0}$, where we will extract $\boldsymbol{A}_B^{-1}\boldsymbol{b}$ from the table as our $\boldsymbol{x}_B$.

2.4 Finding An Initial Basic Feasible Solution

Recall that to start the simplex method algorithm, we need to initialise $\boldsymbol{x}_0$ to be some basic feasible solution in the feasible region. However, this may not be straight-forward. Consider the standard linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} & f(\boldsymbol{x}) \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

for some $\boldsymbol{A} \in \mathbb{R}^{m\times n}$.

We can choose the index set B such that $\boldsymbol{A}_B$ is invertible and so $\boldsymbol{A}^{-1}\boldsymbol{b}$ is a basic solution. However, we might not be so lucky that this basic solution is feasible, i.e., it is not easy to ensure that $\boldsymbol{A}^{-1}\boldsymbol{b} \geq \mathbf{0}$. To address this issue, we consider an *auxiliary linear program* as follows:

First, note that we can change $\boldsymbol{A}$ by multiplying some of its rows by -1 so that $\boldsymbol{b} \geq \mathbf{0}$. Then, we will introduce a new variable $\boldsymbol{y} \in \mathbb{R}^m$ and consider the linear program

$$\begin{aligned} \min_{\boldsymbol{y}\in\mathbb{R}^m} \sum_{i=1}^{m} y_i& \\ \text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} &= \boldsymbol{b} \\ \boldsymbol{x} &\geq \mathbf{0} \\ \boldsymbol{y} &\geq \mathbf{0}. \end{aligned}$$

Now, it is obvious that $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{y} = \boldsymbol{b}$ is a basic feasible solution to the problem. Solving this auxiliary minimisation problem will try to force $\boldsymbol{y} = \mathbf{0}$. If this is possible, i.e., the optimal value of the auxiliary program is 0, then it is clear that we have found some $\boldsymbol{x} \geq \mathbf{0}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, which is exactly a basic feasible solution to our original program!

Technique 2.4.1 ▸ Two-Phase Method

Consider a standard linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}& \\ \text{s.t. } \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{x} &\geq \mathbf{0}. \end{aligned}$$

The two-phase method solves the linear program with the following procedures:

1. Multiplying some rows of $\boldsymbol{A}$ by -1 wherever necessary such that $\boldsymbol{b} \geq \mathbf{0}$.
2. Construct the auxiliary linear program

$$\begin{aligned} \min_{\boldsymbol{y}\in\mathbb{R}^m} \sum_{i=1}^{m} y_i& \\ \text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} &= \boldsymbol{b} \\ \boldsymbol{x} &\geq \mathbf{0} \\ \boldsymbol{y} &\geq \mathbf{0}. \end{aligned}$$

3. Run simplex method on the auxiliary linear program to obtain its optimal solution $(\boldsymbol{y}^*, \boldsymbol{x}^*)$ and optimal value v^*.
4. If $v^* > 0$:
    - The original problem has empty feasible region.

5. Otherwise, $v^* = 0$:
   - Run simplex method on the original problem with $\boldsymbol{x}_0 = \boldsymbol{x}^*$.

Note that in the two-phase method, we need to solve 2 linear programs. This can be computationally expensive, so we wish to find a way to combine the two phases. Here we use the idea of a penalty term. Consider the linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \ & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + M\sum_{i=1}^{m} y_i \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{aligned}$$

where $M > 0$ is an arbitrarily large constant. We can see that if this augmented objective function has a finite optimal value, then it will also force $\boldsymbol{y} = \boldsymbol{0}$. The corresponding $\boldsymbol{x}$ will be the optimal solution of the original problem.

Technique 2.4.2 ▸ Big-M Method

Consider a standard linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \ & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

The big-M method solves the linear program with the following procedures:

1. Multiplying some rows of $\boldsymbol{A}$ by -1 wherever necessary such that $\boldsymbol{b} \geq \boldsymbol{0}$.
2. Augment the original problem with some arbitrarily large constant $M > 0$ into

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \ & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + M\sum_{i=1}^{m} y_i \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

3. Run simplex method on the augmented linear program.
4. If an optimal solution $(\boldsymbol{y}^*, \boldsymbol{x}^*)$ can be found with $\boldsymbol{y}^* = \boldsymbol{0}$:
   - $\boldsymbol{x}^*$ is an optimal solution to the original problem.
5. Otherwise:

- The original problem has empty feasible set or is unbounded.

2.5 Special Cases

Next, we discuss a few special cases when using the simplex method. First, recall that in Definition 2.1.6, we know that a feasible basic solution $\boldsymbol{x}$ is degenerate if there is some redundant active constraint. However, note that this implies that more than one basic variables can leave. Algebraically, this means that there are different $\ell_1, \ell_2 \in B$ such that

$$\bar{\theta}_j = \min\left\{-\frac{x_i}{d_i^j} : i \in B, d_i^j < 0\right\} = -\frac{x_{\ell_1}}{d_{\ell_1}^j} = -\frac{x_{\ell_2}}{d_{\ell_2}^j}.$$

Without loss of generality, suppose we take x_{ℓ_1} as the leaving variable and try to update x_{ℓ_2} to x'_{ℓ_2}. By Proposition 2.2.5, we have

$$x'_{\ell_2} = x_{\ell_2} + \bar{\theta}_j d_{\ell_2}^j = x_{\ell_2} - \frac{x_{\ell_1}}{d_{\ell_1}^j} d_{\ell_2}^j = x_{\ell_2} - \frac{x_{\ell_2}}{d_{\ell_2}^j} d_{\ell_2}^j = 0,$$

so x_{ℓ_2} is indeed degenerate by Definition 2.2.6.

Note also that in some cases, we may have more than one optimal solution for the linear program. Intuitively, this occurs when at an optimal basic feasible solution, we can still find some feasible direction along which the objective value does not deteriorate. Therefore, **an alternative optimal solution can be found if and only if some reduced cost** $\bar{c}_j = 0$ at an optimal solution. There are 2 specific cases:

1. If we can find optimal solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^k$, then the set of optimal solutions is bounded and can be constructed as

$$\operatorname{conv}\left\{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^k\right\}.$$

2. If at an optimal solution $\boldsymbol{x}^*$, there is $\bar{c}_j = 0$ but $\boldsymbol{d}^j \geq \boldsymbol{0}$, it means that $\boldsymbol{x}^* + \theta \boldsymbol{d}^j \geq \boldsymbol{0}$ is feasible for any $\theta \geq 0$, so the set of optimal solutions given by

$$\left\{\boldsymbol{x}^* + \theta \boldsymbol{d}^j : \theta \geq 0\right\}$$

is unbounded.

Similar to case 2 above, if at any basic feasible solution $\boldsymbol{x}^*$, there is some $j \in N$ such that $\bar{c}_j < 0$ and $\boldsymbol{d}^j \geq \boldsymbol{0}$, then any $\boldsymbol{x}^* + \theta \boldsymbol{d}^j$ where $\theta > 0$ is feasible and improves the objective value. This implies that the optimal value can be **improved indefinitely** and so the program is unbounded.

Lastly, suppose we have some linear program with an empty feasible set. We have seen in Section 2.4 that we can use either the two-phase method or the big-M method to detect this, because if the feasible set were to be empty, we will not be able to find an optimal solution for the auxiliary problem such that $\boldsymbol{y}^* = \mathbf{0}$.

3

Duality Theory

3.1 The Dual Problem

A *dual problem* can be seen as an alternative formulation of some linear program which is known as the *primal problem*. The motivation for the dual problem comes from the following observation: suppose we have a well-ordered set S and we wish to find $\inf S$. If it is difficult to minimise S directly, we may first consider a "relaxed" lower bound t, i.e., fix some t such that $t \leq s$ for all $s \in S$.

Then, by collecting all such lower bounds t into a set denoted as T, we know that for all $t \in T$, $t \leq s$ for all $s \in S$. Now, by **maximising** t, we obtain a **greatest possible lower bound** for S. If we can formulate the set T such that $S \cap T \neq \varnothing$, i.e., there is no x such that $\sup T < x < \inf S$, then this greatest lower bound is exactly $\inf S$. Consider the linear program

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \ & \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

The constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ can be re-written as $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. Now, we augment a "penalty" term $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$ to the original objective function for some $\boldsymbol{p} \in \mathbb{R}^m$.

Proposition 3.1.1 ▸ Lower Bound for Optimal Value

Let $\boldsymbol{x}^*$ be an optimal solution to the linear program

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \ & \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} \\ \textit{s.t. } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

Define

$$g(\boldsymbol{p}) := \min_{\boldsymbol{x} \geq \boldsymbol{0}} \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}),$$

then $g(\boldsymbol{p}) \leq \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^*$.

Proof. Since $\boldsymbol{x}^* \geq \boldsymbol{0}$, we have

$$g(\boldsymbol{p}) = \min_{\boldsymbol{x}\geq\boldsymbol{0}} \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})\right) \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^* + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*).$$

Notice that $\boldsymbol{x}^*$ is feasible to the original problem, so $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{0}$. Therefore,

$$g(\boldsymbol{p}) \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^* + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*.$$

□

Proposition 3.1.1 demonstrates a construction for a function g with respect to every linear program (P) such that g always bounds the objective value of (P) below.

Definition 3.1.2 ▶ Dual Problem

For any linear program

$$\begin{aligned} (P) \qquad & \min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ & \text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \quad \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

its **dual problem** is defined as

$$(D) \qquad \max_{\boldsymbol{p}\in\mathbb{R}^m} \min_{\boldsymbol{x}\geq\boldsymbol{0}} \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})\right),$$

and (P) is known as the **primal problem**.

Remark. We know that $g(\boldsymbol{p}) := \min_{\boldsymbol{x}\geq\boldsymbol{0}} \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})\right)$ is an lower bound for the objective value of (P), so (D) will try to search for the greatest lower bound of the optimal value of (P).

Let us study this lower bound function $g(\boldsymbol{p})$ more closely. Observe that

$$g(\boldsymbol{p}) = \boldsymbol{p}^{\mathrm{T}}\boldsymbol{b} + \min_{\boldsymbol{x}\geq\boldsymbol{0}} \left(\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\right)\boldsymbol{x}.$$

Now we consider 2 cases. If $\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A} \geq \boldsymbol{0}$, then it is clear that $\min_{\boldsymbol{x}\geq\boldsymbol{0}} \left(\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\right)\boldsymbol{x} = \boldsymbol{0}$ by taking $\boldsymbol{x} = \boldsymbol{0}$. Otherwise, $(\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A})_j < 0$ for some $j \in \mathbb{Z}^+$, then $\left(\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\right)\boldsymbol{x}$ is unbounded because clearly for any feasible $\boldsymbol{x}$, we have $\left(\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\right)\left(\boldsymbol{x} + \boldsymbol{e}_j\right) < \left(\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\right)\boldsymbol{x}$ and $\boldsymbol{x} + \boldsymbol{e}_j$ is obviously still feasible.

Therefore, we can essentially reduce $g(\boldsymbol{p})$ to

$$g(\boldsymbol{p}) = \begin{cases} \boldsymbol{p}^\mathrm{T}\boldsymbol{b} & \text{if } \boldsymbol{c}^\mathrm{T} - \boldsymbol{p}^\mathrm{T}\boldsymbol{A} \geq \boldsymbol{0} \\ -\infty & \text{otherwise} \end{cases}.$$

Clearly, in order for us to be able to maximise $g(\boldsymbol{p})$, we need $\boldsymbol{c}^\mathrm{T} - \boldsymbol{p}^\mathrm{T}\boldsymbol{A} \geq \boldsymbol{0}$ to be satisfied by the dual problem. This means that we can have the following systematic construction for the dual problem:

Technique 3.1.3 ▸ Formulation of the Dual Problem

Intuitively, if we try to formulate the dual problem of the dual problem, then we should just return to the primal problem.

Proposition 3.1.4 ▸ Dual of the Dual Is the Primal

Let (P) be a primal problem with the dual problem (D). If (D') is the dual problem of (D), then (D') is equivalent to (D).

3.2 Duality Theorems

We see that a valid dual problem is such that $\boldsymbol{p}^\mathrm{T}\boldsymbol{A} \leq \boldsymbol{c}^\mathrm{T}$.

Theorem 3.2.1 ▸ Weak Duality

Let

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \ & \boldsymbol{c}^\mathrm{T}\boldsymbol{x} \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

be a primal problem and consider its dual

$$\begin{aligned} \max_{\boldsymbol{p}\in\mathbb{R}^m} \ & \boldsymbol{p}^\mathrm{T}\boldsymbol{b} \\ \text{s.t. } & \boldsymbol{p}^\mathrm{T}\boldsymbol{A} \leq \boldsymbol{c}^\mathrm{T}, \end{aligned}$$

then

$$\sup \boldsymbol{p}^\mathrm{T}\boldsymbol{b} \leq \inf \boldsymbol{c}^\mathrm{T}\boldsymbol{x}.$$

Proof. Let $\boldsymbol{p}$ and $\boldsymbol{x}$ be feasible, then $\boldsymbol{p}^{\mathrm{T}}\boldsymbol{b} = \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$, so $\sup \boldsymbol{p}^{\mathrm{T}}\boldsymbol{b} \leq \inf \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$. □

The above theorem justifies that the maximum of the dual objective is indeed an lower bound for the primal optimal value. We wish the primal and the dual problems to have an equal optimal value, so that solving one problem is equivalent to the other.

Corollary 3.2.2 ▶ Necessary Condition for Equal Optimal Values

Let $\boldsymbol{x}^*$ and $\boldsymbol{p}^*$ be feasible solutions to a primal problem (P) and its dual problem (D). If $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^* = (\boldsymbol{p}^*)^{\mathrm{T}}\,\boldsymbol{b}$, then $\boldsymbol{x}^*$ and $\boldsymbol{p}^*$ are the optimal solutions.

Another corollary helps us determine the boundedness and feasibility of a problem.

Corollary 3.2.3 ▶ Primal Is Unbounded If and Only If Dual Is Infeasible

A primal problem (P) is unbounded if and only if its dual problem (D) is infeasible.

The opposite statement of the above corollary also holds, i.e., the primal is infeasible if and only if its dual is unbounded.

In fact, all linear programs also gets for free *strong duality*.

Theorem 3.2.4 ▶ Strong Duality

If an linear program has an optimal solution, then so does its dual. Both the primal and the dual problems have the same optimal value.

4

Sensitivity Analysis

4.1 Sensitivity

Suppose that we have found an optimal solution $\boldsymbol{x}^*$ for some linear program, we are interested to know if the original problem is altered, how far away will $\boldsymbol{x}^*$ become from the new optimal solution. Specifically, we want to examine the sensitivity of

1. feasibility $\boldsymbol{A}_B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$, and
2. optimality $\boldsymbol{c}^{\mathrm{T}} - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A} \geq \boldsymbol{0}$.

Suppose $\boldsymbol{b}$ is changed to $\boldsymbol{b} + \delta\boldsymbol{e}_i$. Note that the optimality condition is independent of $\boldsymbol{b}$ and so is unaffected. If $\boldsymbol{x}^*$ were still to be feasible, we must have

$$\boldsymbol{0} \leq \boldsymbol{A}_B^{-1}(\boldsymbol{b} + \delta\boldsymbol{e}_i) = \boldsymbol{x}_B^* + \delta\left(\boldsymbol{A}_B^{-1}\boldsymbol{e}_i\right).$$

The above inequality yields a range of values for δ such that $\boldsymbol{x}^*$ remains optimal and feasible in the altered problem. The optimal value in the altered problem is given by

$$\begin{aligned}\boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}(\boldsymbol{b} + \delta\boldsymbol{e}_i) &= \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{b} + \delta\boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{e}_i \\ &= \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{b} + \delta\left(\boldsymbol{p}^*\right)^{\mathrm{T}}\boldsymbol{e}_i \\ &= \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{b} + \delta p_i^*.\end{aligned}$$

We see that the change in objective value is closely tied to the dual optimal solution, such that a small change of δ in b_i results in a change of δp_i^* in the optimal value.

Definition 4.1.1 ▸ Marginal Cost

Let (P) be a primal problem with dual problem (D). If the dual problem has an optimal solution $\boldsymbol{p}*$, then p^* is called the **marginal cost** or **shadow cost** of b_i for minimisation (P), and the **marginal profit** or **shadow price** of b_i for maximisation (P).

Suppose that $\boldsymbol{c}$ is changed to $\boldsymbol{c} + \delta\boldsymbol{e}_j$ for some j. Note that the feasibility condition is independent of $\boldsymbol{c}$. If x_j is non-basic, the altered reduced cost is

$$\bar{c}_j{}' = c_j + \delta - \boldsymbol{c}_B^{\mathrm{T}}\boldsymbol{A}_B^{-1}\boldsymbol{A} = \bar{c}_j + \delta.$$

Clearly, optimality remains if and only if $\delta \geq -\bar{c_j}$. If x_j is basic, then for each $i \in N$, we need

$$\begin{aligned} c_i - \left(\boldsymbol{c}_B + \delta \boldsymbol{e}_j\right)^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i &= c_i - \boldsymbol{c}_B^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i - \delta \boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i \\ &= \bar{c_j} - \delta \boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i \\ &\geq 0. \end{aligned}$$

If $\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i > 0$, we have $\delta \leq \frac{\bar{c_j}}{\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i}$. If $\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i < 0$, we have $\delta \geq \frac{\bar{c_j}}{\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i}$. If $\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i = 0$, it is clear that $\bar{c_j}$ is unchanged. Therefore, let $\bar{a}_{i,j} = \boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{A}_B^{-1} \boldsymbol{A}_i$, we have

$$\max_{\bar{a}_{i,j} < 0} \frac{\bar{c_j}}{\bar{a}_{i,j}} \leq \delta \leq \min_{\bar{a}_{i,j} > 0} \frac{\bar{c_j}}{\bar{a}_{i,j}}.$$