HKU MATH3901 Study Notes

Chiu Ka Long (Leo)[*]

Last Updated: 2025-05-24

Contents

[*] email ✉: `leockl@connect.hku.hk`; personal website 🔗: `https://leochiukl.github.io`

1 Introduction to Linear Programming

1.0.1 MATH3901 is about *linear programming* (LP), a special type of optimization problem which, as you would expect, involves "linear" things only. While only linearity is allowed in LP, it is still quite flexible and can be applied in many real-life situations, e.g., factory productions 🏭, military plannings 💣 (this is indeed a major motivation for developments in LP!), etc.

1.0.2 Due to the linear nature, linear programming problems are well-studied and many efficient approaches for solving them have already been developed. A natural question is then, *why* are we still studying linear programming when many performant LP solvers are available (e.g., in common coding languages like Python)? A possible answer is that while computer 🖥 can assist you to solve LP problems efficiently, the responsibility of *formulating the right LP problem* ("asking the right question") and *using the tools properly* (easier said than done!) still lies upon the user (you!), so it is still important to understand the theoretical aspects of linear programming and how the algorithms for solving LPs work.

Because of that, instead of focusing on what computers can do very well (performing the often tedious computations for solving LPs), here we will emphasize more on the *conceptual* and *theoretical* aspects of LPs and the study of *algorithms* for solving LPs. [Note: But here we will still solve some relatively "simple" LP problems that require few computations, to better understand the concepts and algorithms.]

1.1 Basic Concepts in Linear Programming

1.1.1 Of course, the first thing we should do is to *define* what a linear programming problem is. As suggested earlier, it is an **optimization problem**, which can be expressed as follows in general:

$$\begin{aligned} \text{min/max} \quad & f(\boldsymbol{x}) \\ \text{subject to (s.t.)} \quad & \boldsymbol{x} = (x_1, \dots, x_n) \in S \end{aligned}$$

where:

- $f : S \to \mathbb{R}$ is the **objective function**;
- $S \subseteq \mathbb{R}^n$ is the **feasible region**, and every $\boldsymbol{x} \in S$ is called a **feasible solution** or **feasible point**;
- $f^* = \inf / \sup\{f(\boldsymbol{x}) : \boldsymbol{x} \in S\}$ is the **optimal value** (always unique, if exists), and *every* $\boldsymbol{x}^* \in S$ with $f(\boldsymbol{x}^*) = f^*$ is called an **optimal solution** (not unique in general!). [Note: We use $\inf / \sup$ rather than $\min / \max$ in the definition of optimal value to allow for more flexibility, e.g., for a minimization problem, if the objective function is not bounded below on S, "inf" would give $-\infty$ while "min" would not exist. Note that in such case, while we have a "$-\infty$" optimal value, no optimal solution actually exists.]

An optimization problem is said to be **feasible** if the feasible region is nonempty. The variables $x_1, \dots, x_n$ are called **decision variables**. Decision variables that are not involved in any constraints (coming from feasible region) are known as **free variables**. The primary goal is to *solve* optimization problems, i.e., finding optimal solutions together with the optimal value (if exists).

All vectors in MATH3901 are assumed to be *column vectors* unless otherwise specified. For convenience in notations, we shall denote the column vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T$ by $(x_1, \dots, x_n)$ if the entries are to be emphasized.

1.1.2 **Defining LP problems.** Here we shall only focus on the special case where (i) the objective function f is <u>linear</u> and (ii) the feasible region S is described by *finitely many* <u>linear</u> equalities or (weak) inequalities[1], which is known as a **linear programming problem**:

$$\begin{aligned} \text{min/max} \quad & f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} = c_1x_1 + \cdots + c_nx_n \\ \text{s.t.} \quad & \boldsymbol{a}_j^T\boldsymbol{x} \le / = / \ge b_j \text{ for all } j = 1, \dots, m \end{aligned}$$

[1] Allowing *infinitely many* linear equalities or inequalities would give us so much flexibility that even *nonlinear* constraints can be described!

where $\boldsymbol{c}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^n$ and $b_1, \ldots, b_m \in \mathbb{R}$. Here the feasible region can be expressed as $S = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_j^T \boldsymbol{x} \leq / = / \geq b_j$ for all $j = 1, \ldots, m\}$. [Note: For discussions on general optimization problems, see MATH3904.]

1.1.3 **Examples of LP problems.**

(1)

$$\begin{aligned} \min \quad & 2x_1 + 3x_2 - x_3 \\ \text{s.t.} \quad & x_1 + 2x_4 \leq 3 \\ & x_2 + x_3 + x_4 = 1 \\ & x_1 + x_3 - 2x_4 \geq -2 \\ & x_1, x_2, x_4 \geq 0 \\ & x_3 \leq 0 \end{aligned}$$

[Note: Here we have $\boldsymbol{c} = (2, 3, -1, 0)$, $\boldsymbol{a}_1 = (1, 0, 0, 2)$, $b_1 = 3$, etc.]

(2)

$$\begin{aligned} \min \quad & x_1 - x_2 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 1 \\ & 2x_2 - 3x_3 = 4 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

We can see that the second example is somewhat "simple" as the only inequality constraints are for nonnegativity of x_1, x_2, and x_3, which are quite "simple" and the rest are only *equality* constraints. On the other hand, the first one involves some more complicated inequality constraints and thus appears to be harder to deal with. In fact, the second LP problem is in the so-called *standard form*, a "nice" form of LP problem that is easy to solve.

1.1.4 **Standard form of LP problems.** A LP problem is in standard form if it takes the following form:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

where $\boldsymbol{c} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ (an $m \times n$ matrix with real entries), and $\boldsymbol{b} \in \mathbb{R}^m$.

Remarks:

- Here, $\boldsymbol{x} \geq \boldsymbol{0}$ carries the componentwise meaning, i.e., $x_1, \ldots, x_n \geq 0$.
- $A\boldsymbol{x} = \boldsymbol{b}$ collects all possible *equality* constraints on the decision variables $x_1, \ldots, x_n$. To see this, we can identify the jth row of A as $\boldsymbol{a}_j^T$, and write $\boldsymbol{b} = (b_1, \ldots, b_m)$, Then, $A\boldsymbol{x} = \boldsymbol{b}$ just refers to the following system of linear equations:

$$\begin{cases} \boldsymbol{a}_1^T \boldsymbol{x} = b_1 \\ \boldsymbol{a}_2^T \boldsymbol{x} = b_2 \\ \qquad \vdots \\ \boldsymbol{a}_m^T \boldsymbol{x} = b_m \end{cases}$$

1.1.5 **Equivalence between optimization problems.** While not every LP is in standard form (of course!), it turns out that it is always possible to *transform* a LP problem into a LP problem *in standard form* such that the two LP problems are *equivalent*.

In general, two optimization problems of minimization[2] are said to be equivalent if given an *arbitrary* feasible solution to *each* problem, one can construct a feasible solution to the *other* system, *with the*

[2] By adding a negative sign to the objective function if necessary, we can always convert a maximization problem to a minimization problem. Hence it suffices to focus on optimization problems of minimization.

same objective value. [Note: Sometimes, "equivalent" refers to the following weaker notion: Given an arbitrary optimal solution to each problem, one can construct an optimal solution to the other system, with the same optimal value. Often this type of equivalence is enough for the purpose of solving optimization problems.]

More specifically, to show that two optimization problems (A) and (B) are equivalent, we need to carry out the following two-step process:

(1) *(A) → (B):* Fix any feasible solution $\boldsymbol{x}_A$ to (A). Based on it, construct a solution $\boldsymbol{x}'_A = u(\boldsymbol{x}_A)$ such that:

   i. $\boldsymbol{x}'_A$ is a feasible solution to (B).

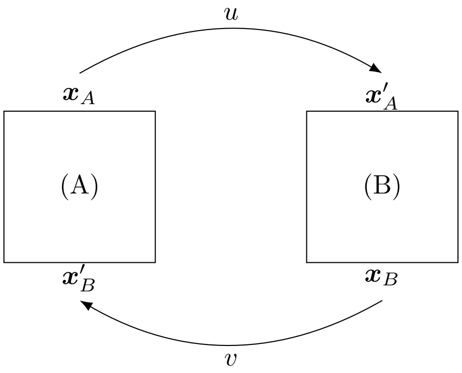
   ii. $\text{obj}_{(A)}(\boldsymbol{x}_A) = \text{obj}_{(B)}(\boldsymbol{x}'_A)$ where $\text{obj}_{(A)}$ and $\text{obj}_{(B)}$ are the objective functions for (A) and (B) respectively.

(2) *(B) → (A):* Fix any feasible solution $\boldsymbol{x}_B$ to (B). Based on it, construct a solution $\boldsymbol{x}'_B = v(\boldsymbol{x}_B)$ such that:

   i. $\boldsymbol{x}'_B$ is a feasible solution to (A).

   ii. $\text{obj}_{(A)}(\boldsymbol{x}'_B) = \text{obj}_{(B)}(\boldsymbol{x}_B)$.

We can then see that the key 🔑 for showing the equivalence is to figure out what the mappings u and v should be.



To understand this better, let us consider a simple example about proving the equivalence between the following two LP problems (A) and (B):

(A):

$$\begin{aligned} \min \quad & x_1 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

(B):

$$\begin{aligned} \min \quad & x_1 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Proof. Fix any feasible solution $\boldsymbol{x} = (a, b)$ to (A). Then, consider the solution $(x_1, x_2, x_3) = (a, b, 1 - a - b)$ to (B). It is feasible for (B) because $1 - a - b \geq 0$ (and $a, b \geq 0$ follows from the feasibility of solution for (A)), and $x_1 + x_2 + x_3 = a + b + (1 - a - b) = 1$. Also, as these two solutions have the same first entry, they have the same objective value a in either LP problem.

Now fix any feasible solution $\boldsymbol{x} = (a, b, c)$ to (B). It is easy to see that the solution $(x_1, x_2) = (a, b)$ is feasible for (A), since $a + b \leq a + b + c = 1$ (as $c \geq 0$) and $a, b \geq 0$. Besides, these two solutions again have the same first entry, hence same objective value. □

1.1.6 Same optimal values for two equivalent optimization problems.

Proposition 1.1.a. Two equivalent optimization problems of minimizations always have the same optimal values.

Proof. First, let us express the two optimization problems as:
(P):

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A \end{aligned}$$

(Q):

$$\begin{aligned} \min \quad & g(\boldsymbol{y}) \\ \text{s.t.} \quad & \boldsymbol{y} \in B \end{aligned}$$

Let f^* and g^* denote the optimal values of (P) and (Q) respectively, with $\boldsymbol{x}^* \in A$ and $\boldsymbol{y}^* \in B$ being corresponding optimal solutions (not necessarily unique). Then we can write $f^* = f(\boldsymbol{x}^*)$ and $g^* = g(\boldsymbol{y}^*)$. We will prove $f^* = g^*$ by showing (i) $f^* \leq g^*$ and (ii) $g^* \leq f^*$.

- $\underline{f^* \leq g^*}$: As $\boldsymbol{y}^* \in B$, by the equivalence, there is a mapping v such that $v(\boldsymbol{y}^*) \in A$ and $f(v(\boldsymbol{y}^*)) = g(\boldsymbol{y}^*) = g^*$. Thus, $f^* = f(\boldsymbol{x}^*) \overset{(\boldsymbol{x}^* \text{ optimal})}{\leq} f(v(\boldsymbol{y}^*)) = g^*$.
- $\underline{g^* \leq f^*}$: As $\boldsymbol{x}^* \in A$, by the equivalence, there is a mapping u such that $u(\boldsymbol{x}^*) \in B$ and $g(u(\boldsymbol{x}^*)) = f(\boldsymbol{x}^*) = f^*$. Thus, $g^* = g(\boldsymbol{y}^*) \overset{(\boldsymbol{y}^* \text{ optimal})}{\leq} g(u(\boldsymbol{x}^*)) = f^*$.

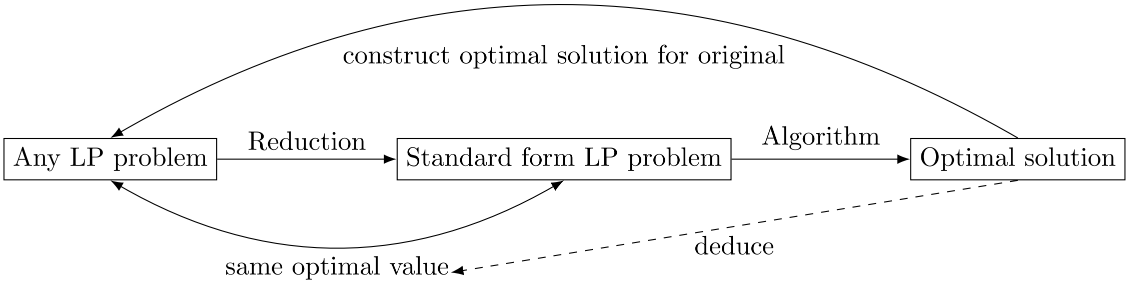
□

[Note: We can see that the weaker notion of "equivalence" suggested earlier (concerning only optimal solutions) is indeed enough for guaranteeing the equality between optimal values. This also explains why this weaker notion is often already enough for many situations.]

1.1.7 Reducing LP problems to standard form.

Reducing LP problems to standard form. Proposition 1.1.a suggests a handy approach to solve LP problem, namely by converting/reducing it to an *equivalent* standard form LP problem, and solve the latter problem instead of the original, which can be done easily in general (many efficient algorithms are available). This approach allows us to:

- obtain the optimal value (if exists) readily from the latter LP problem (by Proposition 1.1.a);
- obtain also optimal solutions in the original LP problem, by constructing from optimal solutions in the latter LP problem (by equivalence).



The following theorem proposes one systematic method for converting any LP problem to a LP problem in standard form.

Theorem 1.1.b. Given any LP problem, consider the following steps.

(1) *Changing to minimization:* If necessary, convert maximization to minimization by changing the objective function $f \to -f$.

(2) *Eliminating inequality constraints:* For every inequality constraint, introduce a **slack variable** (for "$\leq$" case)/**surplus variable** (for "$\geq$" case) $s_j \geq 0$ as follows:

- *("$\leq$" case)* Change $\sum_{i=1}^{n} a_{ji}x_i \leq b_j \to (\sum_{i=1}^{n} a_{ji}x_i) + s_j = b_j$.
- *("$\geq$" case)* Change $\sum_{i=1}^{n} a_{ji}x_i \geq b_j \to (\sum_{i=1}^{n} a_{ji}x_i) - s_j = b_j$.

(3) *Eliminating free variables:* Change every free variable $x_i \to x_i^+ - x_i^-$, where $x_i^+ = \begin{cases} x_i & \text{if } x_i > 0, \\ 0 & \text{if } x_i \leq 0 \end{cases}$ and $x_i^- = \begin{cases} 0 & \text{if } x_i \geq 0, \\ -x_i & \text{if } x_i < 0 \end{cases}$ denote the **positive part** and **negative part** of x_i respectively.

Performing these steps on the given LP problem would yield a LP problem in *standard form* and *equivalent* to the one from original, or obtained after the first step (changing to minimization).

Proof. Omitted; see below for an example that illustrates the main idea. □

[Note: After performing the steps as suggested in Theorem 1.1.b, we would get a LP problem of *minimization* with only equality constraints (except the inequality constraints about nonnegativity of decision variables), and *nonnegative* decision variables coming from three sources: (i) original x_i's, (ii) x_i^+'s and x_i^-'s, and (iii) slack variables s_j's. Thus the resulting LP problem is in standard form.]

1.1.8 **Example of reducing LP problems to standard form.** Consider the following LP problem:

$$\begin{aligned} \max \quad & -x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 3 \\ & 3x_1 + 2x_2 = 14 \\ & x_1 \geq 0 \end{aligned}$$

Now we carry out the steps as indicated in Theorem 1.1.b:

(1) *Changing to minimization:* Change $-x_1 + 3x_2 \to x_1 - 3x_2$ and max → min. (Call the resulting LP problem (A).)

(2) *Eliminating inequality constraints:* Introduce a surplus variable $x_3 \geq 0$ (notation does not matter), and change $x_1 + x_2 \geq 3 \to x_1 + x_2 - x_3 = 3$ (and also add an inequality constraint $x_3 \geq 0$).

(3) *Eliminating free variables:* The only free variable is x_2. So we change *every* $x_2 \to x_2^+ - x_2^-$ (and add inequality constraints $x_2^+, x_2^- \geq 0$).

The resulting LP problem in standard form (call it (B)) is then:

$$\begin{aligned} \min \quad & x_1 - 3x_2^+ + 3x_2^- \\ \text{s.t.} \quad & x_1 + x_2^+ - x_2^- - x_3 = 3 \\ & 3x_1 + 2x_2^+ - 2x_2^- = 14 \\ & x_1, x_2^+, x_2^-, x_3 \geq 0 \end{aligned}$$

While the number of decision variables goes up from 2 to 4, we have successfully simplified the LP problem so that it can be solved by efficient algorithms coded in computer programs.

Here, let us demonstrate how to show the LP problems (A) and (B) are equivalent, providing some insights on the proof of Theorem 1.1.b.

Proof. First fix any feasible solution $(x_1, x_2) = (a, b)$ to (A). Then, consider the solution $(x_1, x_2^+, x_2^-, x_3) = (a, b^+, b^-, 3 - a - (b^+ - b^-))$. It is straightforward to check that this solution is feasible for (B) and has the same objective value (because $b \equiv b^+ - b^-$).

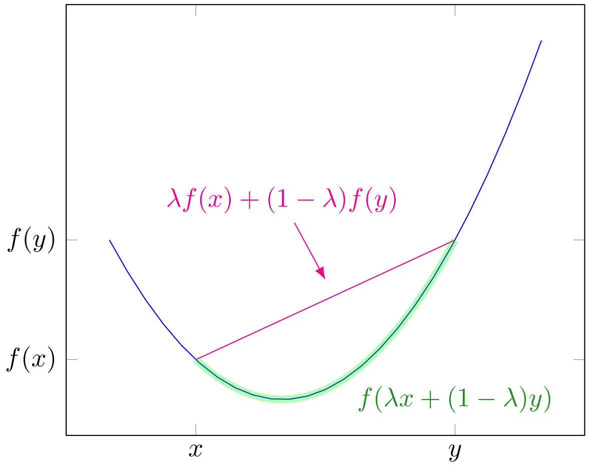
Now fix any feasible solution $(x_1, x_2^+, x_2^-, x_3) = (a, b, c, d)$ to (B). Then, consider the solution $(x_1, x_2) = (a, b - c)$. Again, it is straightforward to check that this solution is feasible for (A) and has the same objective value. □

1.1.9 **Reducing optimization problems to LP problems.** While many optimization problems are not LP problems, we can indeed reduce some of them to LP problems, allowing us to solve them efficiently (through further reduction to standard form, say). One notable example is minimization problems involving *piecewise linear convex* objective function and "≤" constraints.

Preliminaries about convex functions.

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

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


$\lambda f(x) + (1 - \lambda) f(y)$
$f(y)$
$f(x)$
$f(\lambda x + (1 - \lambda) y)$
x
y

- *Property of convex functions:*

Proposition 1.1.c. Let $f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ be convex functions. Then, their maximum $f(x) = \max_{i=1,\ldots,m} f_i(\boldsymbol{x})$ is convex.

Proof. Fix any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and any $\lambda \in [0, 1]$. Then,

$$\begin{aligned} f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) &= \max_{i=1,\ldots,m} f_i(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \\ &\leq \max_{i=1,\ldots,m} \lambda f_i(\boldsymbol{x}) + (1 - \lambda) f_i(\boldsymbol{y}) \\ &\leq \lambda \underbrace{\max_{i=1,\ldots,m} f_i(\boldsymbol{x})}_{f(\boldsymbol{x})} + (1 - \lambda) \underbrace{\max_{i=1,\ldots,m} f_i(\boldsymbol{y})}_{f(\boldsymbol{y})}. \end{aligned}$$

□

Proposition 1.1.c leads us to call a function of the form $f(\boldsymbol{x}) = \max_{i=1,\ldots,m}(\boldsymbol{c}_i^T \boldsymbol{x})$ **piecewise linear convex function**.

Now, consider a minimization problem problem with piecewise linear objective function and "≤" constraint, (A):

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) = \max_{i=1,\ldots,m} \boldsymbol{c}_i^T \boldsymbol{x} \\ \text{s.t.} \quad & \max_{j=1,\ldots,m} \boldsymbol{a}_j^T \boldsymbol{x} \leq b \end{aligned}$$

where $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^n$ and $b \in \mathbb{R}$, with $\boldsymbol{x} = (x_1, \ldots, x_n)$. [Note: Noting that $|x| = \max\{x, -x\}$, this form of optimization problems includes those "LP-like" problems but with absolute value signs.] Of course, the problem (A) itself is not a LP problem in general. But we claim that the problem (A) is indeed equivalent to the following LP problem (B) in the optimal case (i.e., here we are referring to the weaker notion of equivalence concerning only optimal solutions):

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & y \geq \boldsymbol{c}_i^T \boldsymbol{x} \text{ for all } i = 1, \dots, m \\ & \boldsymbol{a}_j^T \boldsymbol{x} \leq b \text{ for all } j = 1, \dots, m \end{aligned}$$

[Note: In the LP problem (B), we have $n+1$ decision variables: $x_1, \dots, x_n, y$. Here we also assume optimal solutions exist.]

Proof. Fix any optimal solution $(x_1, \dots, x_n) = (d_1, \dots, d_n) =: \boldsymbol{d}$ to (A). Then, consider the solution $(x_1, \dots, x_n, y) = (d_1, \dots, d_n, \max_{i=1,\dots,m} \boldsymbol{c}_i^T \boldsymbol{d})$. It is straightforward to check that this solution is also optimal to (B), and it has the same optimal value.

Next, fix any optimal solution $(x_1, \dots, x_n, y) = (d_1, \dots, d_n, d^*)$ to (B). Write $\boldsymbol{d} = (d_1, \dots, d_n)$. The optimality of solution forces that $d^* = \max_{i=1,\dots,m} \boldsymbol{c}_i^T \boldsymbol{d}$, as the latter is the *smallest* value y that satisfies $y \geq \boldsymbol{c}_i^T \boldsymbol{d}$ for every $i = 1, \dots, m$. This means the optimal value for (B) is precisely $\max_{i=1,\dots,m} \boldsymbol{c}_i^T \boldsymbol{d}$.

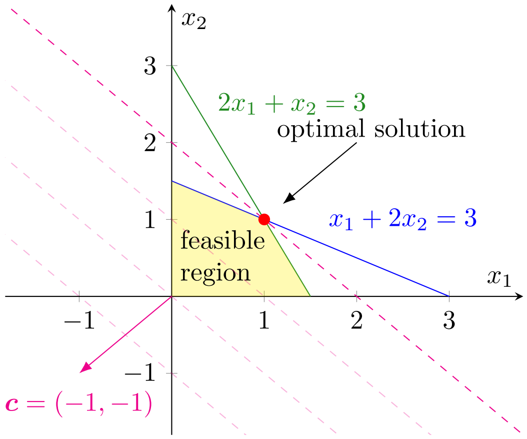
We can then see that the solution $(x_1, \dots, x_n) = (d_1, \dots, d_n)$ is optimal to (A) and has the same optimal value. □

1.2 Geometry of Linear Programming

1.2.1 Apart from analyzing LP problems in algebraic ways like what we did in Section 1.1, another useful method is to consider LP problems *geometrically*, which can yield some valuable insights that cannot be obtained in the algebraic approach. Often, a combination of algebraic and geometric insights is helpful for solving LP problems! Indeed, you may have already learnt a simple geometrical approach for solving LP problems in high school, that involves some "movements of lines"; let us illustrate that approach using the following example.

1.2.2 **"High school" way of solving LP problems.** Consider the following LP problem:

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Graphically, we can "see" 👁 that the optimal solution is $(x_1, x_2) = (1, 1)$ (and it is indeed the case). Let us explain a little bit more about what we are trying to do in the picture above. The idea is to first express the objective function as $\boldsymbol{c}^T \boldsymbol{x}$ where $\boldsymbol{c} = (-1, -1)$ and $\boldsymbol{x} = (x_1, x_2)$. Then, we plot a line

$-x_1 - x_2 = d$ for some constant d, and "move" it around. As the line is "moved" along the direction of $\boldsymbol{c}$, the value of d would rise. Since the LP problem is a *minimization* problem, we would like to "move" the line along the direction *opposite to* $\boldsymbol{c}$ as far as possible.

We can then observe that, when we move the line to the location where $d = 2$, it intersects with the feasible region only at a single point ●. This intersection point $(x_1, x_2) = (1, 1)$ is indeed the optimal solution since, graphically, there would no longer be any feasible solution on the line after moving it even further (i.e., there is not any feasible solution that would lead to an even smaller objective).

⚠ **Limitations.** While this approach is intuitively appealing and simple to carry out (that's why it is often learnt in high school!), a major limitation is that when the number of variables gets larger, it is almost impossible to "visualize" a LP problem in this fashion and solving a LP problem in this way becomes prohibitively difficult!

Nevertheless, solving LP problems "smartly" often *requires* some geometrical insights and ideas. Hence, we will explore the geometrical aspects of LP in Section 1.2.

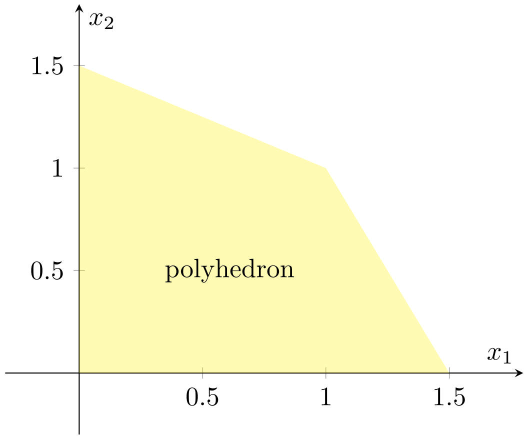
Firstly, we are going to define some geometry terms that describe various elements appearing in the picture that illustrates LP above.

1.2.3 **Polyhedra.** The concept of *polyhedra* is used for describing feasible regions. A set $S \subseteq \mathbb{R}^n$ is a **polyhedron** if $S = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$ for some $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$.

In the example above, the feasible region is

$$\begin{aligned} S &= \{(x_1, x_2) \in \mathbb{R}^2 : x_1 + 2x_2 \leq 3, 2x_1 + x_2 \leq 3, x_1 \geq 0, x_2 \geq 0\} \\ &= \{(x_1, x_2) \in \mathbb{R}^2 : -x_1 + -x_2 \geq -3, -2x_1 - x_2 \geq -3, x_1 \geq 0, x_2 \geq 0\} \\ &= \left\{ (x_1, x_2) \in \mathbb{R}^2 : \begin{bmatrix} -1 & -2 \\ -2 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} -3 \\ -3 \\ 0 \\ 0 \end{bmatrix} \right\}, \end{aligned}$$

which is a polyhedron.

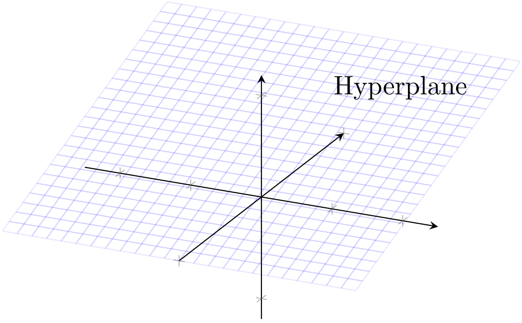


In general, the feasible region of *every* LP problem is a polyhedron, because we can always express each equality/inequality constraint in the "$\geq$" form:

- $\boldsymbol{a}_j^T \boldsymbol{x} \geq b_j$ ✓
- $\boldsymbol{a}_j^T \boldsymbol{x} \leq b_j \to -\boldsymbol{a}_j^T \boldsymbol{x} \geq -b_j$
- $\boldsymbol{a}_j^T \boldsymbol{x} = b_j \to \begin{cases} \boldsymbol{a}_j^T \boldsymbol{x} \geq b_j \\ -\boldsymbol{a}_j^T \boldsymbol{x} \geq -b_j \end{cases}$

1.2.4 **Special cases of polyhedra.** The following types of polyhedra only involve *one* equality/inequality in the condition.

- A set $S \subseteq \mathbb{R}^n$ is a **hyperplane** if $S = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} = \boldsymbol{b}\}$ for some $\boldsymbol{a} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ and $\boldsymbol{b} \in \mathbb{R}$.
- A set $S \subseteq \mathbb{R}^n$ is a **halfspace** if $S = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \geq \boldsymbol{b}\}$ for some $\boldsymbol{a} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ and $\boldsymbol{b} \in \mathbb{R}$.

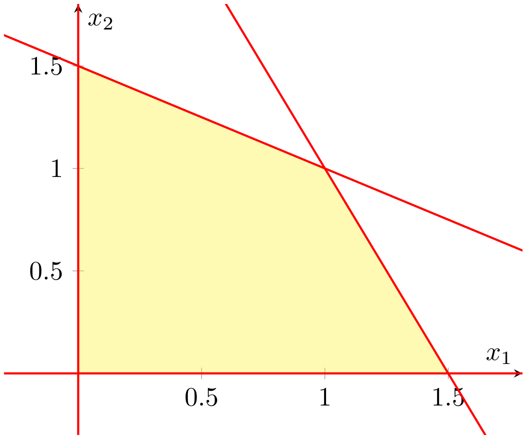


In general, a hyperplane "splits" the Euclidean space $\mathbb{R}^n$ into two parts, and each of them is a *halfspace* (roughly: half of the Euclidean space $\mathbb{R}^n$).

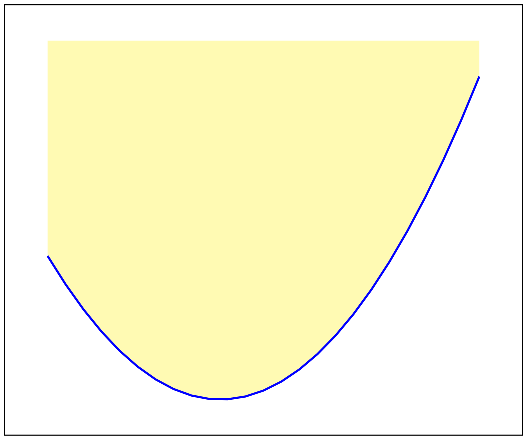
Since a polyhedron P is formed by finitely many "$\geq$" constraints, we can always express it as the *intersection* of finitely many halfspaces:

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_j^T\boldsymbol{x} \geq b_j \text{ for all } j = 1, \ldots, m\} = \bigcap_{j=1}^{m} \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_j^T\boldsymbol{x} \geq b_j\}.$$

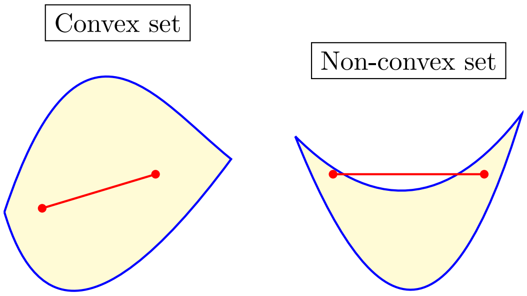
For example, the polyhedron in the previous example is the intersection of four halfspaces (which four?).



1.2.5 **Convex sets.** Geometrically, we can observe that a polyhedron is always "bowed outward". Mathematically, we can describe this feature via *convex set*. A set $S \subseteq \mathbb{R}^n$ is **convex** if for all $\boldsymbol{x}, \boldsymbol{y} \in S$ and all $\lambda \in [0, 1]$, we have $\lambda\boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in S$. It is not hard to see that this definition is rather similar to that for *convex function*. Indeed, a convex function can be defined as a function whose **epigraph** (i.e., the set of points lying on or above the graph of the function) is a convex set.



Graphically speaking, a convex set contains every line segment between two points in the set:



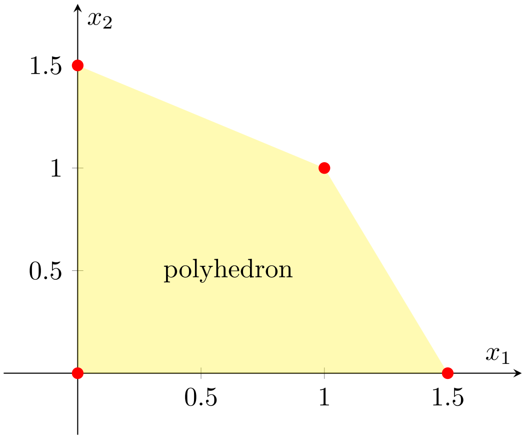
1.2.6 Agreeing with our geometrical intuition, a polyhedron is indeed convex.

Proof. Consider any polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq b\}$. Fix any $\boldsymbol{x}, \boldsymbol{y} \in P$ and any $\lambda \in [0, 1]$. Since

$$A(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) = \lambda A\boldsymbol{x} + (1-\lambda)A\boldsymbol{y} \geq \lambda A\boldsymbol{b} + (1-\lambda)A\boldsymbol{b} = A\boldsymbol{b},$$

we have $\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in P$, hence P is convex. □

1.2.7 **Vertices.** After investigating how to describe the "bowing out" feature of polyhedron mathematically, we will look into another geometrical feature of a polyhedron, namely the appearance of "corners".



For example, for the polyhedron above, intuitively it should have four "corners" or *vertices*, located at the red dots in the picture; they are some "extreme points" or "tips" of the polyhedron. But how should we describe them mathematically?

A rather clever and tricky way to characterize the concept of *vertex* is as follows. Let P be a polyhedron. Then a point $\boldsymbol{x} \in P$ is said to be a **vertex** of P if for all $\boldsymbol{y}, \boldsymbol{z} \in P \setminus \{\boldsymbol{x}\}$ and all $\lambda \in [0,1]$, we have $\boldsymbol{x} \neq \lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{z}$.

In words, a point $\boldsymbol{x}$ is vertex if it is *not* contained in the line segment between any two points that are both different from x. With this definition, those red dots above are indeed vertices of the polyhedron, aligning with our geometrical intuition (check!).

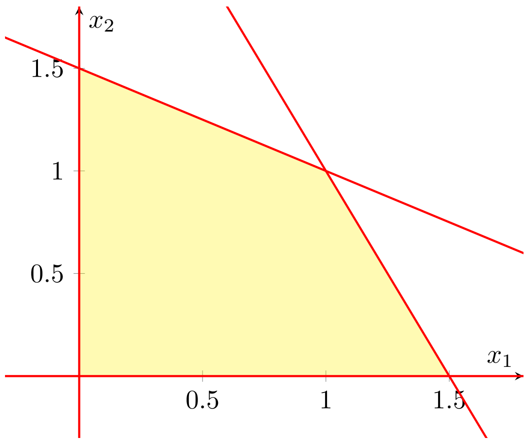
1.2.8 **Algebraic characterization of vertices.** While the above definition of *vertex* is technically "correct", it is rather inconvenient work with, because it is not clear how we can *find* vertices based on *algebraic* expressions of polyhedra! Thus we would like to characterize vertices *algebraically*.

Consider a polyhedron $P \subseteq \mathbb{R}^n$ with the following general expression:

$$P = \{x \in \mathbb{R}^n : \boldsymbol{a}_i^T \boldsymbol{x} \geq b_i,\ \forall i \in M_1; \boldsymbol{a}_i^T \boldsymbol{x} \leq b_i,\ \forall i \in M_2; \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in M_3\}$$

where M_1, M_2, and M_3 are finite index sets. The key 🔑 for characterizing vertices algebraically is to use the idea of *active constraints*. A constraint $\boldsymbol{a}_i^T \boldsymbol{x} \leq / = / \geq b_i$ is **active at $\boldsymbol{x}^*$** if *equality* is achieved at $\boldsymbol{x}^*$, i.e., $\boldsymbol{a}_i^T \boldsymbol{x}^* = b_i$.

In the following picture, points at which some constraints are active are lying on red lines. Our geometrical intuition then tells us that vertices are precisely those points where *many* constraints are active. But what is meant by "many"? To be more precise, we need to use the concept of *basic solution*.



1.2.9 **Basic (feasible) solutions.** Let $\boldsymbol{x}^* \in \mathbb{R}^n$. Then $\boldsymbol{x}^*$ is a **basic solution** (of P) if (i) all equality constraints are active at $\boldsymbol{x}^*$, and (ii) there are n *linearly independent* constraints active at $\boldsymbol{x}^*$.[3] [Note: The constraints $\boldsymbol{a}_{i_1}^T \boldsymbol{x} \geq / = / \leq b_{i_1}, \ldots, \boldsymbol{a}_{i_n}^T \boldsymbol{x} \geq / = / \leq b_{i_n}$ are said to be linearly independent if $\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_n}$ are linearly independent.]

We also say that $\boldsymbol{x}^*$ is a **basic feasible solution** (of P) if it is a basic solution that satisfies all the constraints for P (i.e., is feasible).

[Note: Recall from linear algebra that, for an $n \times n$ matrix A, the following are equivalent:

(a) Rows of A are linearly independent.

(b) Columns of A are linearly independent.

(c) $\operatorname{rank}(A) = n$.

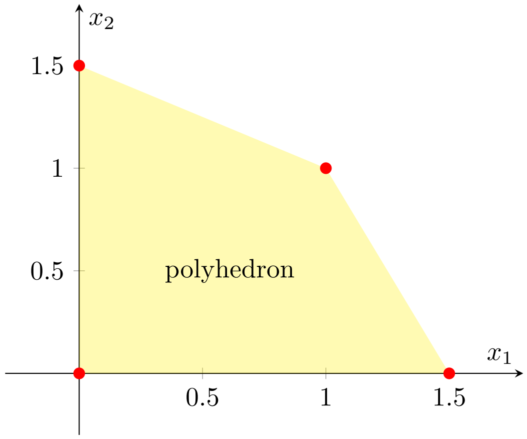
(d) $\det(A) \neq 0$.

[3] We require linear independence to avoid having several "redundant" active constraints, like $x_1^* + x_2^* + x_3^* = 1$ and $2x_1^* + 2x_2^* + 2x_3^* = 2$, which are technically "different" but redundant.

(e) A is invertible.

]

The red points below are indeed all the basic feasible solutions of P, which coincides with our usual understanding of vertices. This is in fact not a coincidence, and being a basic feasible solution is actually *equivalent* to being a vertex.



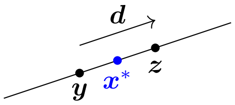
1.2.10 **Equivalent characterization of vertices.**

Theorem 1.2.a (Equivalent characterization of vertices)**.** Let $P \subseteq \mathbb{R}^n$ be a polyhedron and $\boldsymbol{x}^* \in \mathbb{R}^n$. Then $\boldsymbol{x}^*$ is a vertex of P iff $\boldsymbol{x}^*$ is a basic feasible solution of P.

Proof. "⇒": Fix any vertex $\boldsymbol{x}^*$ of P. By definition, $\boldsymbol{x}^* \in P$ so it is feasible. Hence it suffices to show that it is a basic solution of P as well.

Introducing the set of indices for all the active constraints and direction $\boldsymbol{d}$. Due to the feasibility, all equality constraints have to be active at $\boldsymbol{x}^*$, so it remains to show that there are n linearly independent constraints active at $\boldsymbol{x}^*$. Let $I := \{i : \boldsymbol{a}_i^T\boldsymbol{x}^* = b_i\}$ denote the set of indices for all the constraints active at $\boldsymbol{x}^*$. Assume to the contrary that $\operatorname{span}(\{a_i : i \in I\}) \subsetneq \mathbb{R}^n$. Then there is a nonzero vector $\boldsymbol{d} \in \mathbb{R}^n$ such that $\boldsymbol{a}_i^T\boldsymbol{d} = 0$ for all $i \in I$ (it originates from the orthogonal complement $\operatorname{span}(\{a_i : i \in I\})^\perp$, which is nonzero as $\operatorname{span}(\{a_i : i \in I\}) \subsetneq \mathbb{R}^n$).

Moving along the direction $\boldsymbol{d}$ to yield contradiction. Next, we would like to "move along" the direction $\boldsymbol{d}$ to get two points $\boldsymbol{y}, \boldsymbol{z} \in P \setminus \{\boldsymbol{x}^*\}$ such that $\boldsymbol{x}^* = \lambda\boldsymbol{y} + (1-\lambda)\boldsymbol{z}$ for some $\lambda \in [0,1]$, leading to a contradiction. To do that, we choose a sufficiently small $\varepsilon > 0$, and take $\boldsymbol{y} = \boldsymbol{x}^* + \varepsilon\boldsymbol{d}$ and $\boldsymbol{z} = \boldsymbol{x}^* - \varepsilon\boldsymbol{d}$.



To see why these choices achieve what we want, consider:

- $\underline{\boldsymbol{y}, \boldsymbol{z} \in P \setminus \{\boldsymbol{x}^*\}}$: First it is clear that $\boldsymbol{y} \neq \boldsymbol{x}^*$ and $\boldsymbol{z} \neq \boldsymbol{x}^*$ as $\boldsymbol{d} \neq \boldsymbol{0}$. Then consider two cases:
  (a) Fix any $i \in I$. We have $\boldsymbol{a}_i^T\boldsymbol{y} = \underbrace{\boldsymbol{a}_i^T\boldsymbol{x}^*}_{b_i} + \varepsilon\underbrace{\boldsymbol{a}_i^T\boldsymbol{d}}_{0} = b_i$, so the constraint ("="/"≤"/"≥") is satisfied by $\boldsymbol{y}$.
  (b) Fix any $i \notin I$. WLOG assume that the constraint is of the form $\boldsymbol{a}_i^T\boldsymbol{x} \geq b_i$. Since it is not active at $\boldsymbol{x}^*$, we have $\boldsymbol{a}_i^T\boldsymbol{x}^* > b_i$. Thus, $\boldsymbol{a}_i^T\boldsymbol{y} = \boldsymbol{a}_i^T\boldsymbol{x}^* + \varepsilon\boldsymbol{a}_i^T\boldsymbol{d} \overset{(\varepsilon\text{ suff. small})}{>} b_i$. (We can choose ε such that $\varepsilon|\boldsymbol{a}_i^T\boldsymbol{d}| < \boldsymbol{a}_i^T\boldsymbol{x}^* - b_i$.) So the constraint is satisfied by $\boldsymbol{y}$.

The argument is similar for $\boldsymbol{z}$, and so we know $\boldsymbol{y}, \boldsymbol{z} \in P$.

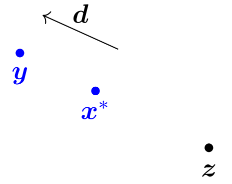
- $\underline{\boldsymbol{x}^* = \lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{z} \text{ for some } \lambda \in [0,1]}$: Note that $\boldsymbol{x}^* = (1/2)\boldsymbol{y} + (1/2)\boldsymbol{z}$.

Contradiction.

Hence we have $\operatorname{span}(\{a_i : i \in I\}) = \mathbb{R}^n$, so from linear algebra we know $\{\boldsymbol{a}_i : i \in I\}$ contains n linearly independent vectors.

"$\Leftarrow$": Fix any basic feasible solution $\boldsymbol{x}^*$ of P, and let $I = \{i : \boldsymbol{a}_i^T \boldsymbol{x}^* = b_i\}$. Assume to the contrary that $\boldsymbol{x}^*$ is not a vertex. Then there exist $\boldsymbol{y}, \boldsymbol{z} \in P \setminus \{\boldsymbol{x}^*\}$ and $\lambda \in [0,1]$ such that $\boldsymbol{x}^* = \lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{z}$.

Introducing a direction $\boldsymbol{d}$ and expressing $\boldsymbol{y}, \boldsymbol{z}$ in terms of it. Let $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x}^* \neq \boldsymbol{0}$. Then $\boldsymbol{y} = \boldsymbol{x}^* + \boldsymbol{d}$ and $\boldsymbol{z} = \boldsymbol{x}^* - t\boldsymbol{d}$ for some $t > 0$.



Obtaining a contradiction by considering $\boldsymbol{a}_i^T \boldsymbol{y}$ and $\boldsymbol{a}_i^T \boldsymbol{z}$. Since $\boldsymbol{x}^*$ is a basic feasible solution of P, $\{\boldsymbol{a}_i : i \in I\}$ contains n linearly independent vectors, thus $\operatorname{span}(\{\boldsymbol{a}_i : i \in I\}) = \mathbb{R}^n$. This means there exists $i \in I$ such that $\boldsymbol{a}_i^T \boldsymbol{d} \neq 0$ (as it is not possible to have a *nonzero* $\boldsymbol{d}$ with $\boldsymbol{a}_i^T \boldsymbol{d} = 0$ $\forall i \in I$). WLOG, assume $\boldsymbol{a}_i^T \boldsymbol{d} > 0$. Then,

$$\boldsymbol{a}_i^T \boldsymbol{y} = \underbrace{\boldsymbol{a}_i^T \boldsymbol{x}^*}_{b_i} + \underbrace{\boldsymbol{a}_i^T \boldsymbol{d}}_{>0} > b_i, \quad \text{while} \quad \boldsymbol{a}_i^T \boldsymbol{z} = \underbrace{\boldsymbol{a}_i^T \boldsymbol{x}^*}_{b_i} - \underbrace{t\boldsymbol{a}_i^T \boldsymbol{d}}_{>0} < b_i.$$

This suggests that it is impossible for $\boldsymbol{y}$ and $\boldsymbol{z}$ to satisfy this constraint simultaneously, thus $\boldsymbol{y} \notin P$ or $\boldsymbol{z} \notin P$, contradiction. □

1.2.11 **Specialized characterization of basic solutions for standard form polyhedra.** For *standard form* polyhedra $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ (corresponding to standard form LP problems), we have a specialized characterization for basic (feasible) solution, which will be helpful in the development of the *simplex method* in Section 2. Recall from [1.2.9] that, for a general polyhedron $P \subseteq \mathbb{R}^n$, $\boldsymbol{x}^* \in \mathbb{R}^n$ is called a basic solution of P if (i) all equality constraints are active at $\boldsymbol{x}^*$, and (ii) there are n *linearly independent* constraints active at $\boldsymbol{x}^*$. Here we will derive an alternative characterization for a *standard form* polyhedron, and the following lemma will be helpful for the proof.

Lemma 1.2.b. Let $\boldsymbol{x}^* \in \mathbb{R}^n$ and let $I = \{i : \boldsymbol{a}_i^T \boldsymbol{x} = b_i\}$ be the set of indices of constraints that are active at $\boldsymbol{x}^*$. Then there are n linearly independent vectors in the set $\{\boldsymbol{a}_i : i \in I\}$ iff the system of equations $\boldsymbol{a}_i^T \boldsymbol{x} = b_i$ $\forall i \in I$ has a unique solution.

Proof. "$\Rightarrow$": Assume there are n linearly independent vectors in the set $\{\boldsymbol{a}_i : i \in I\}$. Then we know that $\operatorname{span}(\{\boldsymbol{a}_i : i \in I\}) = \mathbb{R}^n$. Now assume to the contrary that the system $\boldsymbol{a}_i^T \boldsymbol{x} = b_i$ $\forall i \in I$ has two distinct solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Then for the *nonzero* vector $\boldsymbol{d} = \boldsymbol{x}_1 - \boldsymbol{x}_2$, we have $\boldsymbol{a}_i^T \boldsymbol{d} = 0$ for all $i \in I$, contradicting to $\operatorname{span}(\{\boldsymbol{a}_i : i \in I\}) = \mathbb{R}^n$.

"$\Leftarrow$": Assume the system of equations $\boldsymbol{a}_i^T \boldsymbol{x} = b_i$ $\forall i \in I$ has a unique solution. Due to the uniqueness, it is not possible to have a nonzero vector $\boldsymbol{d}$ with $\boldsymbol{a}_i^T \boldsymbol{d} = 0$ $\forall i \in I$. Thus, the orthogonal complement is $\{\boldsymbol{a}_i : i \in I\}^\perp = \operatorname{span}(\{\boldsymbol{a}_i : i \in I\})^\perp = \{\boldsymbol{0}\}$, which implies that $\operatorname{span}(\{\boldsymbol{a}_i : i \in I\}) = \mathbb{R}^n$. By the reduction approach from linear algebra, we know $\{\boldsymbol{a}_i : i \in I\}$ contains n linearly independent vectors. □

Theorem 1.2.c. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ be a standard form polyhedron, where $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$. Suppose that the m rows of A are linearly independent (which implies that $m \leq n$). Then $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is a basic solution of P iff (i) there exist m linearly independent columns $\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}$ of A such that $\sum_{i=1}^m \boldsymbol{A}_{B(i)} x_{B(i)} = \boldsymbol{b}$, and (ii) $x_{N(1)} = \cdots = x_{N(n-m)} = 0$ where $\{N(1), \ldots, N(n-m)\} = \{1, \ldots, n\} \setminus \{B(1), \ldots, B(m)\}$.

Proof. "$\Leftarrow$": Consider a $\boldsymbol{x} \in \mathbb{R}^n$ and assume that (i) and (ii) hold. Then we have $A\boldsymbol{x} = \sum_{i=1}^{m} \boldsymbol{A}_{B(i)} x_{B(i)} = \boldsymbol{b}$, so the equality constraints are satisfied. Since the vectors $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent, the variables $x_{B(1)}, \dots, x_{B(m)}$ must be uniquely determined: If we have $\sum_{i=1}^{m} \boldsymbol{A}_{B(i)} \widetilde{x}_{B(i)} = \boldsymbol{b}$, then subtracting this from the equation above gives $\sum_{i=1}^{m} \boldsymbol{A}_{B(i)}(x_{B(i)} - \widetilde{x}_{B(i)}) = \boldsymbol{0}$, which implies by the linear independence that $\widetilde{x}_{B(i)} = x_{B(i)}$ for all $i = 1, \dots, m$.

Since the other variables are always zero, the system $A\boldsymbol{x} = \boldsymbol{b}$ has also a unique solution. Thus, by Lemma 1.2.b, we know there are n linearly independent constraints active at $\boldsymbol{x}$.

"$\Rightarrow$": Assume that $\boldsymbol{x} \in \mathbb{R}^n$ is a basic solution. Let $x_{B(1)}, \dots, x_{B(k)}$ be the *only* nonzero variables in $\boldsymbol{x}$, i.e., they are nonzero and $x_i = 0$ for all $i \notin \{B(1), \dots, B(k)\}$. Then all the active equality constraints are given by:

- $\sum_{i=1}^{k} \boldsymbol{A}_{B(i)} x_{B(i)} = \boldsymbol{b}$,
- $x_i = 0$ for all $i \notin \{B(1), \dots, B(k)\}$.

Since there are n linearly independent active ones here, we know that the system of these equations has a unique solution by Lemma 1.2.b. This implies that $\sum_{i=1}^{k} \boldsymbol{A}_{B(i)} x_{B(i)} = \boldsymbol{b}$ has a unique solution. Then, we know that $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(k)}$ are linearly independent; if not, there would be $\lambda_1, \dots, \lambda_k$ that are not all zero such that $\sum_{i=1}^{k} \boldsymbol{A}_{B(i)} \lambda_i = 0$, which would then yield $\sum_{i=1}^{k} \boldsymbol{A}_{B(i)}(x_{B(i)} + \lambda_i) = \boldsymbol{b}$, contradicting to the uniqueness of solution.

Since A has m linearly independent rows, its maximal number of linearly independent columns is also m. Then we deduce that $k \leq m$, and by the extension approach from linear algebra, we can find $m - k$ more columns of A: $\boldsymbol{A}_{B(k+1)}, \dots, \boldsymbol{A}_{B(m)}$, such that $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent.

Lastly, for every $i \in \{N(1), \dots, N(n-m)\}$, we have $i \notin \{B(1), \dots, B(k)\}$ (as $k \leq m$), thus by construction we must have $x_i = 0$. This completes the proof. □

Remarks:

- The variables $x_{B(1)}, \dots, x_{B(m)}$ are called **basic variables**, while the variables $x_{N(1)} \dots, x_{N(n-m)}$ are called **non-basic variables**. The collection $\{\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}\}$ of m linearly independent columns of A is called a **basis** of A, and the matrix formed by these columns, $B = \begin{bmatrix} \boldsymbol{A}_{B(1)} & \cdots & \boldsymbol{A}_{B(m)} \end{bmatrix} \in \mathbb{R}^{m \times m}$, is called a **basis matrix**.
- From the proof of "$\Rightarrow$" direction, we can see that while the non-basic variables have to be zero, the basic variables *could* be zero as well (those $x_{B(k+1)}, \dots, x_{B(m)} = 0$), which would yield *more than* n active constraints at a basic solution $\boldsymbol{x}$, since we already have n active constraints from $A\boldsymbol{x} = \boldsymbol{b}$ (m active constraints) and $x_{N(1)} = \cdots = x_{N(n-m)} = 0$ ($n - m$ active constraints). Such basic solution with more than n active constraints (here this is equivalent to having some basic variables being 0) is called **degenerate**. Geometrically, a degenerate basic solution is at the intersection of three of more lines in two dimensions, or at the intersection of four or more planes in three dimensions, etc.

  It turns out that degeneracy could greatly influence algorithms for LP and cause some undesirable behaviours. As such, in most discussions on the algorithms for LP later, we will assume that every basic feasible solution is *nondegenerate* (so all basic variables $x_{B(1)}, \dots, x_{B(m)}$ are nonzero).
- The assumption that the rows of A are linearly independent is actually not that restrictive. In fact, if the rows of A are linearly *dependent* for a LP problem, it indicates that (i) some equality constraints are redundant and can be removed to make the rows linearly independent without changing the feasible region, or (ii) the feasible region is empty (when some contradictory equality constraints are present), which makes such LP problem not meaningful. Therefore, such assumption can be made throughout the discussion on standard form LP problems without loss of generality.

1.2.12 **Procedure for finding basic feasible solutions of standard form polyhedra.** Based on Theorem 1.2.c, we can obtain the following procedure for finding all basic feasible solutions of any standard form polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ where $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, with rows of A being linearly independent (implying that $m \leq n$):

(1) Choose m linearly independent columns $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ of A.

(2) *(Non-basic variables)* Set $x_i = 0$ for all $i \notin \{B(1), \dots, B(m)\}$.

(3) *(Basic variables)* Solve the system of m equations $A\boldsymbol{x} = B\boldsymbol{x}_B = \boldsymbol{b}$, where $B := \begin{bmatrix}\boldsymbol{A}_{B(1)} & \cdots & \boldsymbol{A}_{B(m)}\end{bmatrix}$ is the basis matrix and $\boldsymbol{x}_B := (x_{B(1)}, \dots, x_{B(m)})$, for the unknowns $x_{B(1)}, \dots, x_{B(m)}$. [Note: Since the columns $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent, the basis matrix B must be invertible, thus the system has a unique solution.]

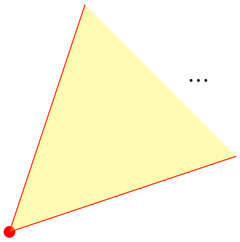
(4) *(Determining feasibility)* If $\boldsymbol{x} \geq \boldsymbol{0}$, then a basic feasible solution is obtained. Otherwise, the basic solution obtained is not feasible.

(5) Repeat (1)-(4) until all combinations of linearly independent columns of A are considered.

From this procedure, since there are at most $\binom{n}{m}$ combinations of linearly independent columns of A, we can deduce that a standard form polyhedron can have at most $\binom{n}{m}$ basic feasible solutions.

1.2.13 **Existence of vertices.** A polyhedron may not have a vertex. For example, hyperplanes and halfspaces do not have vertices. In view of this, we are interested in investigating *when* a polyhedron would have a vertex.

Intuitively, one may think that a *bounded* polyhedron always have a vertex. This is indeed true, but the boundedness is not a necessary condition for the existence of vertex. For example, the following picture illustrates an unbounded polyhedron with a vertex.



It turns out that a *necessary* and *sufficient* condition for existence of vertices is related to whether a *line* (infinitely long one; not “line segment” ⚠) is contained.

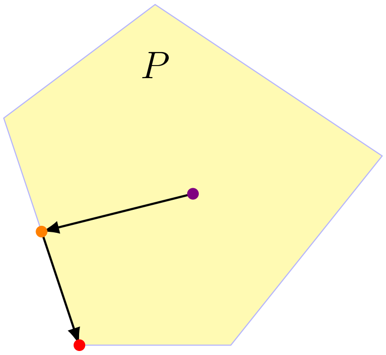
1.2.14 **Necessary and sufficient condition for existence of vertices.** A polyhedron $P \subseteq \mathbb{R}^n$ **contains a line** if there exist $\boldsymbol{x} \in P$ and a nonzero vector $\boldsymbol{d} \in \mathbb{R}^n$ such that $\underbrace{\boldsymbol{x} + \lambda\boldsymbol{d}}_{\text{point on line}} \in P$ for all $\lambda \in \mathbb{R}$.

Theorem 1.2.d (Necessary and sufficient condition for existence of vertices)**.** A nonempty polyhedron P has at least one vertex iff it does *not* contain a line.

Proof. “$\Rightarrow$”: Suppose P has a vertex $\boldsymbol{y}$, which is also a basic feasible solution. Then there are n linearly independent constraints active at $\boldsymbol{y}$, say $\boldsymbol{a}_i^T\boldsymbol{y} = \boldsymbol{b}_i$ for all $i = 1, \dots, n$. Note that there exists $i = 1, \dots, n$ such that for all nonzero vector $\boldsymbol{d} \in \mathbb{R}^n$, $\boldsymbol{a}_i^T\boldsymbol{d} \neq \boldsymbol{0}$; otherwise, the orthogonal complement $\operatorname{span}(\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_n\})^{\perp}$ would be nonzero and we would have $\operatorname{span}(\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_n\}) \subsetneq \mathbb{R}^n$, contradiction.

WLOG, assume the ith constraint is $\boldsymbol{a}_i^T\boldsymbol{x} \geq \boldsymbol{b}_i$. Then, for all $\boldsymbol{x} \in P$ and all nonzero vectors $\boldsymbol{d} \in \mathbb{R}^n$, we can choose λ sufficiently positive/negative (depending on the sign of $\boldsymbol{a}_i^T\boldsymbol{d}$) such that $\boldsymbol{a}_i^T(\boldsymbol{x} + \lambda\boldsymbol{d}) = \boldsymbol{a}_i^T\boldsymbol{x} + \lambda\boldsymbol{a}_i^T\boldsymbol{d} < b_i$, thus $\boldsymbol{x} + \lambda\boldsymbol{d} \notin P$. This means P does not contain a line.

“$\Leftarrow$”: *Idea:* The following picture illustrates the key 🔑 for this part of proof: We start with an arbitrary point in P, and then keep moving along a direction such that one more “edge” is touched (one more constraint becomes active) without leaving P, until reaching a vertex (basic feasible solution).



Introducing a set of indices for active constraints. Assume P does not contain a line. Fix any $\boldsymbol{x} \in P$ and let $I := \{i : \boldsymbol{a}_i^T \boldsymbol{x} = b_i\}$ denote the set of indices for all the constraints active at $\boldsymbol{x}$. If there are n linearly independent constraints active at $\boldsymbol{x}$, then $\boldsymbol{x}$ is by definition a basic feasible solution, so a vertex of P is found.

Moving along a direction to "touch" one more "edge" without leaving P. So, assume henceforth that there are only *less than* n linearly independent constraints active at $\boldsymbol{x}$. In such case, we have $\operatorname{span}(\{\boldsymbol{a}_i : i \in I\}) \subsetneq \mathbb{R}^n$, and thus there is a nonzero vector $\boldsymbol{d} \in \mathbb{R}^n$ such that $\boldsymbol{a}_i^T \boldsymbol{d} = 0$ for all $i \in I$. Then consider the line $\boldsymbol{x} + \lambda \boldsymbol{d}$ where $\lambda \in \mathbb{R}$ is the parameter.

Constraints that are active at $\boldsymbol{x}$ are also active at every point on this line since, for all $i \in I$ and $\lambda \in \mathbb{R}$, we have $\boldsymbol{a}_i^T(\boldsymbol{x} + \lambda \boldsymbol{d}) = \boldsymbol{a}_i^T \boldsymbol{x} + \lambda \boldsymbol{a}_i^T \boldsymbol{d} = \boldsymbol{a}_i^T \boldsymbol{x} + 0 = \boldsymbol{a}_i^T \boldsymbol{x}$. This suggests that moving along the direction $\boldsymbol{d}$ at least would not reduce the number of "edges touched".

Now, we will show that it is possible to "touch" one more "edge" by choosing an appropriate parameter λ. By assumption, this line must not be contained by P, so as we vary the parameter λ, some constraints will eventually be violated. Then consider the moment at which some constraints are about to be violated (for the first time); in such case, there is at least one new constraint becoming active.[4] Denoting the parameter in such case by λ^*, we then have $\boldsymbol{a}_j^T(\boldsymbol{x} + \lambda^* \boldsymbol{d}) = b_j$ for some $j \notin I$ (we have $j \notin I$ as the constraints corresponding to indices in I would still be active, thus not be violated for all λ). By construction, no constraints are actually violated at the point $\boldsymbol{x} + \lambda^* \boldsymbol{d}$ (we just have that some constraints are *about* to be violated), so $\boldsymbol{x} + \lambda^* \boldsymbol{d}$ is feasible and belongs to P still.

Showing the linear independence of constraints with the additional one. To prove that $\{\boldsymbol{a}_i : i \in I\} \cup \{\boldsymbol{a}_j\}$ is linearly independent, it suffices to show that $\boldsymbol{a}_j$ is a not a linear combination of vectors in $\{\boldsymbol{a}_i : i \in I\}$. On one hand, we have $\boldsymbol{a}_j^T \boldsymbol{x} \neq b_j$ since $j \notin I$, and $\boldsymbol{a}_j^T(\boldsymbol{x} + \lambda^* \boldsymbol{d}) = b_j$. This forces $\boldsymbol{a}_j^T \boldsymbol{d} \neq 0$. On the other hand, we know that $\boldsymbol{a}_i^T \boldsymbol{d} = 0$ for all $i \in I$, which implies that $\boldsymbol{a}_j$ is a not a linear combination of vectors in $\{\boldsymbol{a}_i : i \in I\}$; otherwise we would have $\boldsymbol{a}_j^T \boldsymbol{d} = 0$!

Showing the existence of vertices by applying the previous argument iteratively. The argument above depicts a method to raise the number of active constraints through moving to an appropriate point in P. Using this method iteratively, we will eventually reach a point in P at which n linearly independent constraints are active; that point is the vertex we would like to find. □

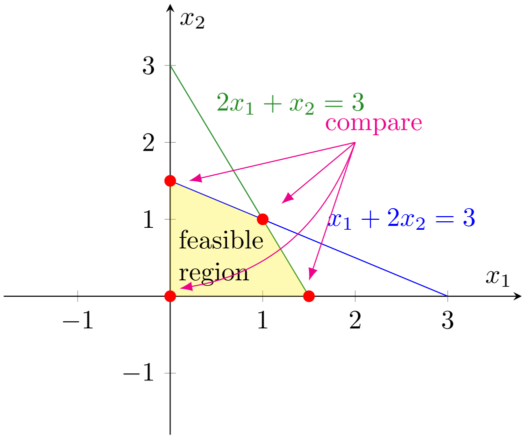
Corollary 1.2.e. Every nonempty standard form polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ has at least one vertex.

Proof. Note that $P \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \geq \boldsymbol{0}\}$ and the latter does not contain a line. Thus P does not contain a line also, so it has at least one vertex by Theorem 1.2.d. □

1.2.15 **A primer of applying geometrical concepts for solving LP problems: optimality of vertices.** So far we have discussed quite a lot of geometrical concepts about LP. But ultimately, as we mention at the start, they are used for solving LP problems. So, to illustrate how they are useful for LP problems, we will study a somewhat simple application of geometrical concepts for solving LP problems, which is about optimality of vertices.

[4] More formally, this can be justified by the intermediate value theorem.

This application may perhaps be learnt in high school as well. In simple terms, it suffices to compare the values of objective functions at all the vertices, and the smallest value is precisely the optimal value (for an minimization problem); this method may be even more straightforward than the previous method which involves "moving lines".



In the following, we will formalize this idea and justify why this method works.

Theorem 1.2.f. Consider a LP problem of minimization:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x} \in P \end{aligned}$$

where the feasible region P is a polyhedron. Suppose P has at least one vertex. Then either the optimal value is $-\infty$[5] or there exists a vertex which is an optimal solution.

Proof. Idea: The proof strategy is similar to the one used for proving the "$\Leftarrow$" direction of Theorem 1.2.d, but we need to ensure additionally that the values taken by the objective function do not increase after moving to new points.

Assume P has at least one vertex. It suffices to show that any finite optimal value is achieved at a vertex of P.

Introducing a set of indices for active constraints. Suppose that the optimal value is finite. Fix any point $\boldsymbol{x} \in P$ at which there are only *less than* n linearly independent constraints that are active. Let $I := \{i : \boldsymbol{a}_i^T \boldsymbol{x} = b_i\}$ denote the set of indices for all the constraints active at $\boldsymbol{x}^*$.

Moving along a direction to "touch" one more "edge" without leaving P or increasing the value taken by objective function. Since there are only *less than* n linearly independent active constraints, we have $\operatorname{span}(\{\boldsymbol{a}_i : i \in I\}) \subsetneq \mathbb{R}^n$, thus there is a nonzero vector $\boldsymbol{d}$ such that $\boldsymbol{a}_i^T \boldsymbol{d} = 0$ for all $i \in I$. By taking negative of $\boldsymbol{d}$ if necessary, we may assume that $\boldsymbol{c}^T \boldsymbol{d} \leq 0$. Now consider two cases:

- *Case 1:* $\boldsymbol{c}^T \boldsymbol{d} < 0$. Consider the *half-line* $\boldsymbol{x} + \lambda \boldsymbol{d}$ where $\lambda \geq 0$ is the parameter. Like the proof for "$\Leftarrow$" direction of Theorem 1.2.d, all the active constraints at $\boldsymbol{x}$ are still active at every point on this half-line, i.e., $\boldsymbol{a}_i^T (\boldsymbol{x} + \lambda \boldsymbol{d}) = 0$ for all $i \in I$.

  Claim: This half-line is not contained in P.

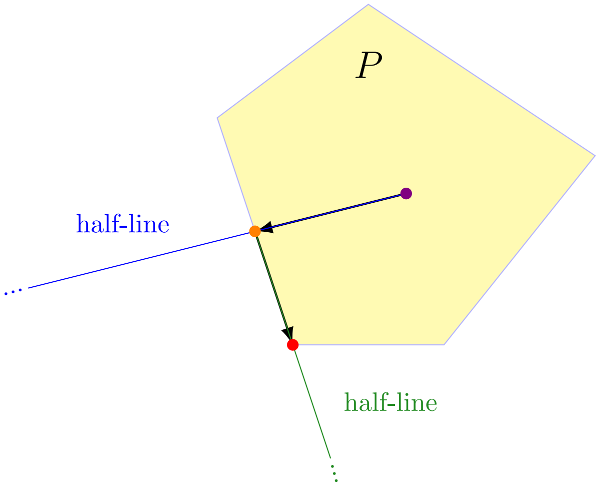
  Proof. If this half-line were contained in P, the objective function would not be bounded below on P, as it can take arbitrarily negative value on this half-line by setting arbitrarily large λ, due to the assumption that $\boldsymbol{c}^T \boldsymbol{d} < 0$. Therefore, the optimal value would be $-\infty$ in such case, which is impossible by our assumption at the beginning. □

  This means that as we increase the parameter $\lambda \geq 0$ for this half-line, some constraints will eventually be violated. Like before, denoting by λ^* the parameter for the moment at which some

[5] This corresponds to the case where the objective function is not bounded below on P.

constraints are *about* to be violated (for the first time), we have $\boldsymbol{a}_j^T(\boldsymbol{x}+\lambda^*\boldsymbol{d}) = b_j$ for some $j \notin I$ and $\boldsymbol{x}+\lambda^*\boldsymbol{d} \in P$. Using a similar argument, we also know that $\{\boldsymbol{a}_i : i \in I\} \cup \{\boldsymbol{a}_j\}$ is linearly independent, increasing the number of linearly independent active constraints by at least one.

In addition, since $\boldsymbol{c}^T\boldsymbol{d} < 0$, we have $\boldsymbol{c}^T(\boldsymbol{x}+\lambda^*\boldsymbol{d}) < \boldsymbol{c}^T\boldsymbol{x}$, meaning that the value taken by the objective function does not increase at this new point.



- *Case 2:* $\boldsymbol{c}^T\boldsymbol{d} = 0$. Consider the line $\boldsymbol{x}+\lambda\boldsymbol{d}$ where $\lambda \in \mathbb{R}$ is the parameter. Since P has at least one vertex, by Theorem 1.2.d P does not contain a line also. Thus, this line must not be contained in P. Again, like the proof for "$\Leftarrow$" direction of Theorem 1.2.d, there exists $\lambda^* \in \mathbb{R}$ such that (i) $\boldsymbol{a}_j^T(\boldsymbol{x}+\lambda^*\boldsymbol{d}) = b_j$ for some $j \notin I$, (ii) $\boldsymbol{x}+\lambda^*\boldsymbol{d} \in P$, and (iii) $\{\boldsymbol{a}_i : i \in I\} \cup \{\boldsymbol{a}_j\}$ is linearly independent. So this again increases the number of linearly independent active constraints by at least one.

  Furthermore, as $\boldsymbol{c}^T\boldsymbol{d} = 0$, we have $\boldsymbol{c}^T(\boldsymbol{x}+\lambda^*\boldsymbol{d}) = \boldsymbol{c}^T\boldsymbol{x}$, meaning that the value taken by the objective function again does not increase at this new point.

Showing that the optimal value is achieved at a vertex by applying the previous argument iteratively. Applying the argument above iteratively, we can find a vertex/basic feasible solution $\boldsymbol{w}_{\boldsymbol{x}} \in P$ such that $\boldsymbol{c}^T\boldsymbol{w}_{\boldsymbol{x}} \le \boldsymbol{c}^T\boldsymbol{x}$. Note that there can only be finitely many vertices/basic feasible solutions of P (as there are only finitely many ways to choose n linearly independent constraints to be active, out of the finitely many constraints in a LP problem), so we may denote them by $\boldsymbol{w}_1, \dots, \boldsymbol{w}_r$.

Let $\boldsymbol{w}^*$ be a vertex of P at which the objective function takes the smallest value among all the vertices (which always exists as there are only finitely many vertices), i.e., $\boldsymbol{c}^T\boldsymbol{w}^* \le \boldsymbol{c}^T\boldsymbol{w}_k$ for all $k = 1, \dots, r$. Note that for all $\boldsymbol{x} \in P$:

- If there are n linearly independent constraints that are active at $\boldsymbol{x}$, then $\boldsymbol{x} = \boldsymbol{w}_k$ for some $k = 1, \dots, r$, so $\boldsymbol{c}^T\boldsymbol{w}^* \le \boldsymbol{c}^T\boldsymbol{w}_k = \boldsymbol{c}^T\boldsymbol{x}$.
- If there are less than n linearly independent constraints that are active at $\boldsymbol{x}$, then by our previous argument there is a vertex $\boldsymbol{w}_k$ such that $\boldsymbol{c}^T\boldsymbol{w}_{\boldsymbol{k}} \le \boldsymbol{c}^T\boldsymbol{x}$ for some $k = 1, \dots, r$. (The vertex $\boldsymbol{w}_x$ must be one of the vertices $\boldsymbol{w}_1, \dots, \boldsymbol{w}_r$.) Thus, $\boldsymbol{c}^T\boldsymbol{w}^* \le \boldsymbol{c}^T\boldsymbol{x}$.

This shows that $\boldsymbol{c}^T\boldsymbol{w}^* \le \boldsymbol{c}^T\boldsymbol{x}$ for all $\boldsymbol{x} \in P$, and hence the optimal value is achieved at the vertex $\boldsymbol{w}^*$. □

Corollary 1.2.g. Consider a LP problem of minimization:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x} \in P \end{aligned}$$

where the feasible region P is a nonempty polyhedron. Then either the optimal value is $-\infty$ or there exists an optimal solution.

Proof. This follows by first reducing the LP problem to an equivalent LP problem in standard form, where the feasible region would be a polyhedron with at least one vertex (by Corollary 1.2.e), and then applying Theorem 1.2.f to such standard form LP problem. □

[Note: Generally, for an optimization problem of minimization, it is possible that the optimal value is not $-\infty$, and there is also no optimal solution. For instance, consider the optimization problem of minimizing $1/x$ subject to $x \geq 1$. The optimal value is $\inf\{1/x : x \geq 1\} = 0$, but there is no optimal solution since $1/x > 0$ for all $x \geq 1$.]

2 The Simplex Method

2.0.1 In [1.1.7], we have mentioned that algorithms are available for solving *standard form* LP problems. A popular one is the *simplex method*, which is the main topic to be discussed in Section 2. Throughout we shall focus on standard form LP problems.

2.1 Optimality Conditions

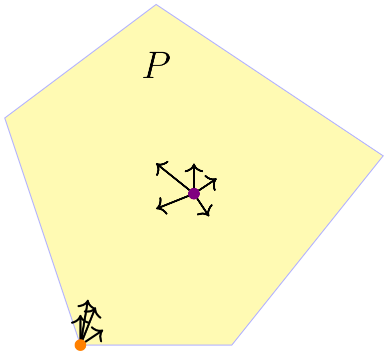
2.1.1 The concept of *optimality conditions* is critical for the simplex method and also many other optimization algorithms. Indeed, to solve basic optimization problems in your previous calculus class, you were also utilizing optimality conditions (e.g., setting derivatives to be zero, etc.), which are usually *necessary* conditions for optimality: If the point in consideration is optimal, then the condition must be satisfied.

Here, we are interested in *sufficient* conditions for optimality instead: If the condition is satisfied, then the point in consideration must be optimal. Those sufficient conditions are very helpful for designing algorithms for solving optimization problems, that utilize the approach of "exploring" different candidates of optimal solutions in the feasible region and checking their optimality through verifying the sufficient condition. The simplex method here is also an algorithm of this type.

Our goal here is to obtain a sufficient condition for optimality that only requires a *finite* amount of computations, so that it can be actually verified in practice and utilized in practical algorithms. (Ultimately, the purpose of designing such algorithms is to implement them in practice for solving optimization problems!)

2.1.2 **Feasible directions.** An useful notion for developing such optimality condition is *feasible direction*. Let $\boldsymbol{x}$ be a point in a polyhedron P. A vector $\boldsymbol{d} \in \mathbb{R}^n$ is called a **feasible direction** at $\boldsymbol{x}$, if there exists $\theta > 0$ such that $\boldsymbol{x} + \theta \boldsymbol{d} \in P$.

Intuitively, feasible direction at $\boldsymbol{x}$ refers to a direction along which the point $\boldsymbol{x}$ can "move" slightly without leaving the polyhedron P (remaining feasible).



With the concept of feasible direction, we can obtain a sufficient and necessary condition for optimality, which can be expressed in words as "moving along any feasible direction would not reduce the objective function value".

Proposition 2.1.a. Consider a LP problem of minimization $\boldsymbol{c}^T\boldsymbol{x}$ over a polyhedron P. A point $\boldsymbol{x}^* \in P$ is optimal iff $\boldsymbol{c}^T\boldsymbol{d} \geq 0$ for all feasible directions $\boldsymbol{d}$ at $\boldsymbol{x}^*$.

Proof. "⇒": We prove by contrapositive. Assume $\boldsymbol{c}^T\boldsymbol{d} < 0$ for some feasible direction $\boldsymbol{d}$ at $\boldsymbol{x}^*$. Then we can move along that direction to further reduce the objective function value, since there exists $\theta > 0$ such that $\boldsymbol{x}^* + \theta\boldsymbol{d} \in P$, and we have $\boldsymbol{c}^T(\boldsymbol{x}^* + \theta\boldsymbol{d}) = \boldsymbol{c}^T\boldsymbol{x} + \theta\boldsymbol{c}^T\boldsymbol{d} < \boldsymbol{c}^T$. Hence $\boldsymbol{x}^*$ is not optimal.

"⇐": Assume $\boldsymbol{c}^T\boldsymbol{d} \geq 0$ for all feasible directions $\boldsymbol{d}$ at $\boldsymbol{x}^*$. Fix any $\boldsymbol{y} \in P$. Since $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x}^*$ is a feasible direction at $\boldsymbol{x}^*$ (consider $\theta = 1$), we have $\boldsymbol{c}^T\boldsymbol{d} \geq 0$, which implies $\boldsymbol{c}^T\boldsymbol{y} \geq \boldsymbol{c}^T\boldsymbol{x}^*$, thus $\boldsymbol{x}^*$ is optimal. □

While Proposition 2.1.a provides a sufficient and necessary condition for optimality, it requires an *infinite* amount of computations (as there are infinitely many feasible directions in general!). So we are not satisfied with this condition and will continue our search 🔍 for a verifiable sufficient condition for optimality.

2.1.3 **Feasible directions at basic feasible solutions.** Observe from the picture above that there are "fewer" feasible directions at basic feasible solutions (vertices), so if we would like to "simplify" the optimality condition in Proposition 2.1.a to a condition that is practically verifiable, it appears that we should focus on optimality condition for basic feasible solutions.

We start by studying the feasible directions at basic feasible solutions.

Proposition 2.1.b. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ be a standard form polyhedron, where $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, with rows of A being linearly independent. Let $\boldsymbol{x}$ be a basic feasible solution of P with the corresponding basis matrix $B = \begin{bmatrix} \boldsymbol{A}_{B(1)} & \cdots & \boldsymbol{A}_{B(m)} \end{bmatrix}$. Write $\boldsymbol{d} = (d_1, \dots, d_n)$, $\boldsymbol{d}_B := (d_{B(1)}, \dots, d_{B(m)})$, $\boldsymbol{d}_N := (d_{N(1)}, \dots, d_{N(n-m)})$, and $N := \begin{bmatrix} \boldsymbol{A}_{N(1)} & \cdots & \boldsymbol{A}_{N(n-m)} \end{bmatrix}$, where $N(1), \dots, N(n-m)$ are the indices corresponding to the non-basic variables.

(a) If $\boldsymbol{d}$ is a feasible direction at $\boldsymbol{x}$, then $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$ and $\boldsymbol{d}_N \geq \boldsymbol{0}$.

(b) If $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$, $\boldsymbol{d}_N \geq \boldsymbol{0}$, *and* $\boldsymbol{x}$ *is nondegenerate*, then $\boldsymbol{d}$ is a feasible direction at $\boldsymbol{x}$.

Proof.

(a) Since $\boldsymbol{d}$ is a feasible direction at $\boldsymbol{x}$, we have $\boldsymbol{x} + \theta\boldsymbol{d} \in P$ for some $\theta > 0$. This means that (i) $A(\boldsymbol{x} + \theta\boldsymbol{d}) = \boldsymbol{b}$ and (ii) $\boldsymbol{x} + \theta\boldsymbol{d} \geq \boldsymbol{0}$. From (i), we have $A\boldsymbol{x} + \theta A\boldsymbol{d} = \boldsymbol{b} \implies \theta A\boldsymbol{d} = \boldsymbol{b} - A\boldsymbol{x} \overset{(\boldsymbol{x}\in P)}{=} \boldsymbol{0} \implies A\boldsymbol{d} = \boldsymbol{0}$. By writing $A\boldsymbol{d} = B\boldsymbol{d}_B + N\boldsymbol{d}_N$, we deduce that $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$.

Next, note that $\boldsymbol{x}_N = \boldsymbol{0}$ by Theorem 1.2.c. Hence, from (ii) we have $\boldsymbol{x}+\theta\boldsymbol{d} \geq \boldsymbol{0} \overset{\text{(only consider non-basic ones)}}{\implies} \boldsymbol{x}_N + \theta\boldsymbol{d}_N \geq \boldsymbol{0} \implies \boldsymbol{d}_N \geq \boldsymbol{0}$.

(b) Assume $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$, $\boldsymbol{d}_N \geq \boldsymbol{0}$, and $\boldsymbol{x}$ is a nondegenerate basic feasible solution of P. Then, we have $A\boldsymbol{d} = B\boldsymbol{d}_B + N\boldsymbol{d}_N = -BB^{-1}N\boldsymbol{d}_N + N\boldsymbol{d}_N = -N\boldsymbol{d}_N + N\boldsymbol{d}_N = \boldsymbol{0}$. Therefore, $A(\boldsymbol{x} + \theta\boldsymbol{d}) = A\boldsymbol{x} + \theta A\boldsymbol{d} = \boldsymbol{b}$ for all $\theta > 0$. So it suffices to find a $\theta > 0$ such that $\boldsymbol{x} + \theta\boldsymbol{d} \geq 0$.

For non-basic variables, we always have $\boldsymbol{x}_N + \theta\boldsymbol{d}_N = \boldsymbol{0} + \theta\boldsymbol{d}_N \geq \boldsymbol{0}$, for every $\theta > 0$. Hence we only need to consider the basic variables. Since $\boldsymbol{x}$ is a nondegenerate basic feasible solution, we must have $x_{B(1)}, \dots, x_{B(m)} > 0$. Therefore, we can choose a sufficiently small $\theta > 0$ such that $\boldsymbol{x}_B + \theta\boldsymbol{d}_B \geq \boldsymbol{0}$. [Note: More precisely, we can choose $\theta = \min_{i=1,\dots,m:d_{B(i)}<0}\{-x_{B(i)}/d_{B(i)}\}$[6] if $d_{B(i)} < 0$ for some $i = 1, \dots, m$, and can choose any $\theta > 0$ if $d_{B(i)} \geq 0$ for all $i = 1, \dots, m$.] Thus, with this $\theta > 0$, we have $\boldsymbol{x} + \theta\boldsymbol{d} \geq \boldsymbol{0}$.

□

2.1.4 **Verifiable optimality condition for basic feasible solutions.** Knowing the properties of feasible directions from Proposition 2.1.b, we can obtain a verifiable optimality condition for basic feasible solutions of standard form polyhedra.

Proposition 2.1.c. Consider a standard form LP problem: minimizing $\boldsymbol{c}^T\boldsymbol{x}$ over a standard form polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$. Let $\boldsymbol{x}$ be a basic feasible solution. Write $\boldsymbol{c} = (c_1, \dots, c_n)$, $\boldsymbol{c}_B := (c_{B(1)}, \dots, c_{B(m)})$, and $\boldsymbol{c}_N := (c_{N(1)}, \dots, c_{N(n-m)})$. For every $j = 1, \dots, n$, let $\bar{c}_j := c_j - \boldsymbol{c}_B^T B^{-1}\boldsymbol{A}_j$ denote the **reduced cost** of x_j (at $\boldsymbol{x}$). Then:

(a) If $\bar{c}_j \geq 0$ for all $j = N(1), \dots, N(n-m)$, then $\boldsymbol{x} \in P$ is optimal.

(b) If $\boldsymbol{x} \in P$ is optimal and *nondegenerate*, then $\bar{c}_j \geq 0$ for all $j = N(1), \dots, N(n-m)$.

Proof.

[6] This expression will appear again later in the development of the simplex method.

(a) Assume that $\bar{c}_j \geq 0$ for all $j = N(1), \ldots, N(n-m)$. Fix any feasible direction $\boldsymbol{d}$ at $\boldsymbol{x}$. Then, by Proposition 2.1.b, we have $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$ and $\boldsymbol{d}_N \geq \boldsymbol{0}$. Hence,

$$\begin{aligned}\boldsymbol{c}^T\boldsymbol{d} &= \boldsymbol{c}_B^T\boldsymbol{d}_B + \boldsymbol{c}_N^T\boldsymbol{d}_N = -\boldsymbol{c}_B^T B^{-1}N\boldsymbol{d}_N + \boldsymbol{c}_N^T\boldsymbol{d}_N \\ &= (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T B^{-1}N)\boldsymbol{d}_N = \sum_{i=1}^{n-m}\left(c_{N(i)} - \boldsymbol{c}_B^T B^{-1}\boldsymbol{A}_{N(i)}\right)d_{N(i)} = \sum_{i=1}^{n-m}\underbrace{\bar{c}_{N(i)}}_{\geq 0}\underbrace{d_{N(i)}}_{\geq 0} \geq 0.\end{aligned}$$

Thus, by Proposition 2.1.a we conclude that $\boldsymbol{x}$ is optimal.

(b) Assume to the contrary that $\boldsymbol{x} \in P$ is optimal and nondegenerate, but $\bar{c}_j < 0$ for some $j = N(1), \ldots, N(n-m)$. Then, by setting $\boldsymbol{d}_j = 1$, $\boldsymbol{d}_{j'} = 0$ all non-basic indices $j' \neq j$, and $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$ (this will appear again in [2.1.5]), we know by Proposition 2.1.b that $\boldsymbol{d}$ is a feasible direction at $\boldsymbol{x}$, so $\boldsymbol{x} + \theta\boldsymbol{d} \in P$ for some $\theta > 0$.

But then by construction we would have $\boldsymbol{c}^T\boldsymbol{d} \overset{\text{(see above)}}{=} \sum_{i=1}^{n-m}\bar{c}_{N(i)}d_{N(i)} = \bar{c}_j < 0$, implying that $\boldsymbol{c}^T(\boldsymbol{x} + \theta d) < \boldsymbol{c}^T\boldsymbol{d}$, contradicting to the optimality of $\boldsymbol{x}$.

□

Remarks:

- This optimality condition is *verifiable* as it only involves checking $n - m$ signs ✓.
- Due to Proposition 2.1.c, a basis matrix B is said to be **optimal** if (i) $\boldsymbol{x}_B = B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$ *(feasibility)* and (ii) $\bar{c}_j \geq 0$ for all $j = N(1), \ldots, N(n-m)$ *(optimality)*. If we have obtained a basic feasible solution with optimal basis matrix, then such basic feasible solution must be optimal. Also, every nondegenerate optimal basic feasible solution must admit an optimal basis matrix.

2.1.5 **More about reduced cost.** To interpret reduced cost, in the equation $\boldsymbol{c}^T\boldsymbol{d} = \sum_{i=1}^{n-m}\bar{c}_{N(i)}\boldsymbol{d}_{N(i)}$ from the proof above, we can set $\boldsymbol{d}_j = 1$, $\boldsymbol{d}_{j'} = 0$ for all non-basic indices $j' \neq j$, and $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{d}_N$. This gives $\boldsymbol{c}^T\boldsymbol{d} = \bar{c}_j$, which suggests that the reduced cost $\bar{c}_j$ of x_j refers to the rate of change $\boldsymbol{c}^T\boldsymbol{d}$ of objective function value (or cost) when moving along such direction $\boldsymbol{d}$; this is the change in cost per unit increase in x_j (namely c_j), reduced by the amount $\boldsymbol{c}_B^T B^{-1}\boldsymbol{A}_j$ (serving as an adjustment for the changes in basic variables from moving along such direction).

With this interpretation, it should be intuitive that the reduced cost of any basic variable x_j is always zero, as the adjustment would intuitively reduce the whole rate of change of cost from the increase in basic variable. It can also be mathematically proved as follows:

Proof. Since x_j is a basic variable, we know $j = B(k)$ for some $k = 1, \ldots, m$. Hence, we have $B^{-1}\boldsymbol{A}_j = B^{-1}\boldsymbol{A}_{B(k)} = \boldsymbol{e}_k$ (note that $I_m = B^{-1}B = B^{-1}\begin{bmatrix}\boldsymbol{A}_{B(1)} & \cdots & \boldsymbol{A}_{B(m)}\end{bmatrix} = \begin{bmatrix}B^{-1}\boldsymbol{A}_{B(1)} & \cdots & B^{-1}\boldsymbol{A}_{B(m)}\end{bmatrix}$). Therefore, the reduced cost is

$$\bar{c}_j = c_j - \boldsymbol{c}_B^T B^{-1}\boldsymbol{A}_j = c_j - \boldsymbol{c}_B^T\boldsymbol{e}_k = c_j - c_{B(k)} = c_j - c_j = 0.$$

□

Because of this, we can rewrite the optimality condition from Proposition 2.1.c in the *row* vector form: $(\bar{c}_1, \ldots, \bar{c}_n)^T =: \bar{\boldsymbol{c}}^T \geq \boldsymbol{0}$, where we have $\bar{\boldsymbol{c}}^T = \boldsymbol{c}^T - \boldsymbol{c}_B^T B^{-1}A$ (here $\boldsymbol{0}$ denotes zero *row* vector). In words, this means a basic feasible solution is optimal if the reduced costs (of all variables) are nonnegative.

2.2 Development and Implementation of the Simplex Method

2.2.1 **Intuition for the simplex method.** Before looking into the details of the simplex method, we first study the main idea behind the simplex method. Consider a standard form LP problem:

$$\begin{aligned}\min\quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.}\quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}\end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$ (with rows being linearly independent) and $\boldsymbol{b} \in \mathbb{R}^m$. Let us collect some previous results about the standard form LP problem below:

(1) (Theorem 1.2.f) If the optimal value is finite, then it must be achieved at one of the basic feasible solutions.

(2) ([1.2.12]) A standard form polyhedron only has finitely many basic feasible solutions.

(3) ([2.1.5]) A basic feasible solution is optimal if the reduced costs are nonnegative.

Based on the results (1) and (2) alone, we can already obtain a straightforward method ("high school approach") for solving standard form LP problems as follows: Assuming that the optimal value is finite, we can compute the objective function value at each of the finitely many basic feasible solutions, and the smallest value is the optimal value, with the corresponding basic feasible solution(s) being the optimal solution(s).

However, this method has some disadvantages 👎:

- *(Assumption on the finiteness of optimal value)* The optimal value is required to be finite, but we may not know whether this is the case before solving the LP problem.
- *(High computational cost)* When the values of m and n are large, there would be a very large number of basic feasible solutions to be considered, leading to a high computational cost.

The *simplex method* to be developed here does not have these disadvantages (though it is less straightforward). Apart from the results (1)-(2), it also utilizes (3), which provides us a sufficient condition for optimality. The key 🔑 idea of the simplex method is to *move between basic feasible solutions until the reduced costs are all nonnegative*, i.e., going through the finitely many basic feasible solutions and checking the optimality of each of them in the process (whether the reduced costs are all nonnegative or not).

2.2.2 **Implementation details for the simplex method.** While the intuitive idea of the simplex method is given above, it is not enough for the actual implementation; we also need to specify how such "movements between basic feasible solutions" should take place. To be more precise, we need to figure out a way to determine both the *direction* and *magnitude* of the next movement in the algorithm, starting from a certain basic feasible solution $\boldsymbol{x}$ at which the reduced costs are not yet all nonnegative, say $\bar{c}_j < 0$ for some non-basic index j. Unless otherwise specified, we shall assume throughout that every basic feasible solution is *nondegenerate*.

For the *direction*, we should naturally move along a feasible direction where the objective function value decreases (as we are solving a minimization problem). Recall from [2.1.5] that we have $\boldsymbol{c}^T\boldsymbol{d} = \bar{c}_j$ if we set $d_j = 1$ and $d_{j'} = 0$ for all non-basic indices $j' \neq j$. Since we have $\bar{c}_j < 0$ here, moving along such direction would lead to a change of $\boldsymbol{c}^T\boldsymbol{d} < 0$ in the objective function value (a decrease), which is what we want. Hence, we will choose the direction $\boldsymbol{d}$ as follows:

- *(Non-basic variables)* $\boxed{d_j = 1}$ and $\boxed{d_{j'} = 0}$ for all non-basic indices $j' \neq j$.
- *(Basic variables)* $\boxed{\boldsymbol{d}_B} = -B^{-1}N\boldsymbol{d}_N \overset{(j=N(i)\text{ for some }i)}{=} -B^{-1}N\boldsymbol{e}_i = -B^{-1}\boldsymbol{A}_{N(i)} \boxed{= -B^{-1}\boldsymbol{A}_j}$

[Note: Under nondegeneracy, this direction $\boldsymbol{d}$ must be a feasible direction by Proposition 2.1.b.]

For the *magnitude*, we are referring to the value of $\theta > 0$ in $\boldsymbol{y} = \boldsymbol{x} + \theta\boldsymbol{d}$ where $\boldsymbol{y}$ is the destination of the movement. Since we have $\boldsymbol{c}^T\boldsymbol{d} < 0$ from above, naturally we would like to set θ as large as possible so that the decrease in the objective function value is maximized. However, it may be possible that $\boldsymbol{y} = \boldsymbol{x} + \theta\boldsymbol{d} \notin P$ if the θ chosen is too large.

Inspired by the proof of Proposition 2.1.b, we set the θ as:

$$\boxed{\theta^* = \min_{i=1,\dots,m:d_{B(i)}<0}\left\{-\frac{x_{B(i)}}{d_{B(i)}}\right\}}$$

if $d_{B(i)} < 0$ for some $i = 1, \dots, m$, which is the largest possible θ to ensure that the basic variables are still nonnegative (thus remaining feasible) in this case.

If we have $d_{B(i)} \geq 0$ for all $i = 1, \dots, m$, it means that $\boldsymbol{x} + \theta \boldsymbol{d}$ remains feasible for all $\theta > 0$, so we can set θ to be arbitrarily large. But in such case, it implies that the objective function value can be arbitrarily negative, and hence the objective function is not bounded below in the feasible region, making the optimal value $-\infty$.

2.2.3 **A change-of-basis point of view of the simplex method.** Following the steps put forward in [2.2.2], assuming $d_{B(i)} < 0$ for some $i = 1, \dots, m$, we will move from the starting basic feasible solution $\boldsymbol{x}$ to another point $\boldsymbol{x}^* = \boldsymbol{x} + \theta^* \boldsymbol{d} \in P$ (which is indeed another basic feasible solution of P, to be shown in Proposition 2.2.a).

Let ℓ be a minimizing index for the formula of θ^*, i.e., $-x_{B(\ell)}/d_{B(\ell)} = \theta^*$, which implies that $x^*_{B(\ell)} = x_{B(\ell)} + \theta^* d_{B(\ell)} = 0$, meaning that the variable with index $B(\ell)$ in $\boldsymbol{x}^*$ becomes zero. On the other hand, since $x_j = 0$ and $d_j = 1$, we have $x_j^* = 0 + \theta^*(1) = \theta^* > 0$. Therefore, after an iteration in the simplex method, (i) a variable with basic index originally ($x^*_{B(\ell)}$) becomes zero and (ii) a variable with nonbasic index originally (x_j^*) becomes positive.

Naturally, this suggests us to replace the column $\boldsymbol{A}_{B(\ell)}$ in the original basis by the column $\boldsymbol{A}_j$, forming a new matrix

$$\begin{aligned}\bar{B} &= \begin{bmatrix} \boldsymbol{A}_{B(1)} & \boldsymbol{A}_{B(\ell-1)} & \boldsymbol{A}_j & \boldsymbol{A}_{B(\ell+1)} & \cdots & \boldsymbol{A}_{B(m)} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_{\bar{B}(1)} & \boldsymbol{A}_{\bar{B}(\ell-1)} & \boldsymbol{A}_{\bar{B}(\ell)} & \boldsymbol{A}_{\bar{B}(\ell+1)} & \cdots & \boldsymbol{A}_{\bar{B}(m)} \end{bmatrix},\end{aligned}$$

where $\bar{B}(i) = B(i)$ for all $i \neq \ell$ and $\bar{B}(\ell) = j$. We say that the column $\boldsymbol{A}_{B(\ell)}$ **exits the basis** and the column $\boldsymbol{A}_j$ **enters the basis** in this case. Let us justify that such matrix $\bar{B}$ is indeed a basis matrix and we can view $\boldsymbol{x}^*$ as a basic feasible solution associated with the basis matrix $\bar{B}$ below.

Proposition 2.2.a.

(a) The columns $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(\ell-1)}, \boldsymbol{A}_j, \boldsymbol{A}_{B(\ell+1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent (hence $\bar{B}$ is a basis matrix).

(b) $\boldsymbol{x}^* = \boldsymbol{x} + \theta^* \boldsymbol{d}$ is a basic feasible solution of P associated with the basis matrix $\bar{B}$.

Proof.

(a) By a result from linear algebra, $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(\ell-1)}, \boldsymbol{A}_j, \boldsymbol{A}_{B(\ell+1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent iff $B^{-1}\boldsymbol{A}_{B(1)}, \dots, B^{-1}\boldsymbol{A}_{B(\ell-1)}, B^{-1}\boldsymbol{A}_j, B^{-1}\boldsymbol{A}_{B(\ell+1)}, \dots, B^{-1}\boldsymbol{A}_{B(m)}$ are linearly independent. So we will prove the latter.

Since $\begin{bmatrix} B^{-1}\boldsymbol{A}_{B(1)} & \cdots & B^{-1}\boldsymbol{A}_{B(m)} \end{bmatrix} = B^{-1}B = I_m$, we know $B^{-1}\boldsymbol{A}_{B(i)} = \boldsymbol{e}_i$ for all $i \neq \ell$, which particularly means that their ℓth entries are all zero.

On the other hand, we have $B^{-1}\boldsymbol{A}_j = -\boldsymbol{d}_B$, whose ℓth entry is $-\boldsymbol{d}_{B(\ell)} > 0$ (as $\boldsymbol{d}_{B(\ell)} < 0$ by the definition of ℓ). This means that $B^{-1}\boldsymbol{A}_j$ is not a linear combination of $B^{-1}\boldsymbol{A}_{B(i)}, \quad i \neq \ell$ (which are linearly independent among themselves). The desired result then follows by the extension approach from linear algebra.

(b) By construction of θ^* and $\boldsymbol{d}$, we have $A\boldsymbol{x}^* = \boldsymbol{b}$ and $\boldsymbol{x}^* \geq \boldsymbol{0}$, so $\boldsymbol{x}^*$ is feasible. Also, for all $i \notin \{\bar{B}(1), \dots, \bar{B}(m)\}$, we have either $i = B(\ell)$ or $i = j'$ where j' is an original non-basic index different from j. In the former case, $x_i^* = x_{B(\ell)} = 0$ as discussed above. In the latter case, $x_i^* = x_{j'}^* = x_{j'} + \theta d_{j'} = x_{j'} + 0 \overset{(x_{j'} \text{ non-basic})}{=} 0$. So $x_i^* = 0$ for all $i \notin \{\bar{B}(1), \dots, \bar{B}(m)\}$, and $\sum_{i=1}^m \boldsymbol{A}_{\bar{B}(i)} x^*_{\bar{B}(i)} = A\boldsymbol{x}^* = \boldsymbol{b}$.

Together with the linear independence $\boldsymbol{A}_{\bar{B}(1)}, \dots, \boldsymbol{A}_{\bar{B}(m)}$ (shown above), we conclude that $\boldsymbol{x}^*$ is a basic feasible solution of P by Theorem 1.2.c, associated with the basis matrix $\bar{B}$.

□

2.2.4 **Summary of the simplex method.** Based on the previous results, we can summarize the procedures carried out in the **simplex method** as follows.

(1) Start with a basic feasible solution $\boldsymbol{x}$ with basis matrix $B = \begin{bmatrix} \boldsymbol{A}_{B(1)} & \cdots & \boldsymbol{A}_{B(m)} \end{bmatrix}$, which is given by $\boldsymbol{x}_B = B^{-1}\boldsymbol{b}$ and $\boldsymbol{x}_N = \boldsymbol{0}$ (see [1.2.12]).

(2) Compute the reduced costs $\bar{c}_j = c_j - \boldsymbol{c}_B^T B^{-1}\boldsymbol{A}_j$ for all non-basic indices j.

  i. If $\bar{c}_j \geq 0$ for all non-basic indices j, then the current basic feasible solution $\boldsymbol{x}$ is optimal, and the algorithm terminates ⏸.

  ii. Otherwise, fix a non-basic index j with $\bar{c}_j < 0$.

(3) Compute $\boldsymbol{u} = B^{-1}\boldsymbol{A}_j$ (which equals $-\boldsymbol{d}_B$).

  i. If no entry of $\boldsymbol{u}$ is positive, then the optimal value is $-\infty$, and the algorithm terminates ⏸.

  ii. If some entries of $\boldsymbol{u}$ are positive, set

$$\theta^* = \min_{i=1,\dots,m: u_i > 0} \left\{ \frac{x_{B(i)}}{u_i} \right\}.$$

(4) Let ℓ be a minimizing index, i.e., $x_{B(\ell)}/u_\ell = \theta^*$. Form a new basis matrix $\bar{B}$ by replacing the column $\boldsymbol{A}_{B(\ell)}$ by $\boldsymbol{A}_j$. For the new basic feasible solution $\boldsymbol{x}^*$, the basic variables are given by $x^*_{B(i)} = x_{B(i)} - \theta^* u_i$ for all $i \neq \ell$ and $x^*_j = \theta^*$ (and the non-basic variables are all zero).

(5) Repeat (1)-(4) with the starting basic feasible solution updated to the new one obtained in the previous iteration, until the algorithm terminates ⏸.

Remarks:

- We do not need to explicitly compute the movement direction $\boldsymbol{d}$ in the process.
- It can be shown that, as long as the feasible region is nonempty and every basic feasible solution is nondegenerate, the algorithm above must terminate after a finite number of iterations.

2.2.5 **Updating B^{-1} through elementary row operations.** In the simplex method depicted in [2.2.4], one major disadvantage is that it requires the computation of B^{-1} in each iteration, which is computationally more costly than the computations of vector inner products $\boldsymbol{a}^T\boldsymbol{b}$, and matrix-vector products $A\boldsymbol{b}$. In view of this deficiency, the *revised* simplex method is developed to avoid excessive computations of matrix inverse in the algorithm. The key 🔑 idea is to just compute B^{-1} once at the initialization, and only *update* B^{-1} in a "smarter" way, that is less costly than recomputing the matrix inverse, in the subsequent iterations.

Such update of B^{-1} is developed based on the observation that, the change in the basis matrix after an iteration is $B \to \bar{B}$, in which only *one* column changes ($\boldsymbol{A}_{B(\ell)} \to \boldsymbol{A}_{\bar{B}(\ell)} = \boldsymbol{A}_j$). As the basis matrix only undergoes a somewhat "small" change, it is natural to expect that the corresponding matrix inverse won't change much also. Indeed, we can change $B^{-1} \to \bar{B}^{-1}$ via performing *elementary row operations* (EROs) only.

From linear algebra, we know that performing EROs corresponds to left-multiplying elementary matrices. So what we want to do now is to find out a sequence of elementary matrices $E_1, \dots, E_r$ such that $E_r \cdots E_1 B^{-1} = \bar{B}^{-1}$ which can be rewritten as $E_r \cdots E_1 B^{-1}\bar{B} = \bar{B}^{-1}\bar{B} = I_m$. This means that it suffices to find a sequence of EROs that turn $B^{-1}\bar{B}$ to I_m, which can be obtained as follows.

Since $\begin{bmatrix} B^{-1}\boldsymbol{A}_{B(1)} & \cdots & B^{-1}\boldsymbol{A}_{B(m)} \end{bmatrix} = B^{-1}B = I_m$, we know $B^{-1}\boldsymbol{A}_{B(\ell)} = \boldsymbol{e}_i$ for all $i \neq \ell$ (you should have seen this in the proof of Proposition 2.2.a already). Thus, we can write

$$\begin{aligned} B^{-1}\bar{B} &= \begin{bmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_{\ell-1} & \boldsymbol{u} & \boldsymbol{e}_{\ell+1} & \cdots & \boldsymbol{e}_m \end{bmatrix} \\ &= \begin{bmatrix} 1 & \cdots & 0 & u_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & u_{\ell-1} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & u_\ell & 0 & \cdots & 0 \\ 0 & \cdots & 0 & u_{\ell+1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & u_m & 1 & \cdots & 1 \end{bmatrix} \end{aligned}$$

[Note: We have $\boldsymbol{u} = -\boldsymbol{d}_B = B^{-1}\boldsymbol{A}_j$.] From this expression, we can observe that the following sequence of EROs can turn $B^{-1}\bar{B}$ to I_m:

(1) *(Changing $u_\ell \to 1$)* First perform the ERO $(1/u_\ell)\boldsymbol{r}_\ell \to \boldsymbol{r}_\ell$. [Note: We have $u_\ell > 0$ by the definition of ℓ.]

(2) *(Zeroizing the ith entry of the column $\boldsymbol{u}$ for all $i \neq \ell$)* For all $i \neq \ell$, perform the ERO $-u_i\boldsymbol{r}_\ell + \boldsymbol{r}_i \to \boldsymbol{r}_i$ (i.e., adding $-u_i$ times the ℓth row to the ith row).

In short, we are adding each of the rows a multiple of the ℓth row to make the column corresponding to $\boldsymbol{u}$ equal the vector $\boldsymbol{e}_\ell$.

2.2.6 **Revised simplex method.** Building upon the idea in [2.2.5], the **revised simplex method** is as follows.

(1) Start with a basic feasible solution $\boldsymbol{x}$ with basis matrix $B = \begin{bmatrix}\boldsymbol{A}_{B(1)} & \cdots & \boldsymbol{A}_{B(m)}\end{bmatrix}$, which is given by $\boldsymbol{x}_B = B^{-1}\boldsymbol{b}$ and $\boldsymbol{x}_N = \boldsymbol{0}$ (see [1.2.12]). At initialization, compute the inverse B^{-1}.

(2) Compute the reduced costs $\bar{c}_j = c_j - \boldsymbol{c}_B^T B^{-1}\boldsymbol{A}_j$ for all non-basic indices j.

  i. If $\bar{c}_j \geq 0$ for all non-basic indices j, then the current basic feasible solution $\boldsymbol{x}$ is optimal, and the algorithm terminates ⏸.

  ii. Otherwise, fix a non-basic index j with $\bar{c}_j < 0$.

(3) Compute $\boldsymbol{u} = B^{-1}\boldsymbol{A}_j$ (which equals $-\boldsymbol{d}_B$). [Note: The reason why we use the vector $\boldsymbol{u}$ instead of $\boldsymbol{d}_B$ here will become transparent later.]

  i. If no entry of $\boldsymbol{u}$ is positive, then the optimal value is $-\infty$, and the algorithm terminates ⏸.

  ii. If some entries of $\boldsymbol{u}$ are positive, set

$$\theta^* = \min_{i=1,\dots,m:u_i>0}\left\{\frac{x_{B(i)}}{u_i}\right\}.$$

(4) Let ℓ be a minimizing index, i.e., $x_{B(\ell)}/u_\ell = \theta^*$. Form a new basis matrix $\bar{B}$ by replacing the column $\boldsymbol{A}_{B(\ell)}$ by $\boldsymbol{A}_j$. For the new basic feasible solution $\boldsymbol{x}^*$, the basic variables are given by $x^*_{B(i)} = x_{B(i)} - \theta^* u_i$ for all $i \neq \ell$ and $x^*_j = \theta^*$ (and the non-basic variables are all zero).

(5) Form an augmented matrix $\left[B^{-1}|\boldsymbol{u}\right] \in \mathbb{R}^{m\times(m+1)}$. Add each of its rows a multiple of the ℓth row to make the last column equal to $\boldsymbol{e}_\ell$. After that, the matrix on the left-hand side would become the inverse of the new basis matrix $\bar{B}^{-1}$.

(6) Repeat (1)-(5) with the starting basic feasible solution and the inverse of the basis matrix updated to the new ones obtained in the previous iteration, until the algorithm terminates ⏸.

2.2.7 **Tableau representation of the simplex method.** While the procedure of the revised simplex method in [2.2.6] can be implemented in computer program </> to solve standard form LP problems efficiently, it is still a bit cumbersome to be carried out *by hand* ✎, due to the numerous quantities to be tracked and computed. Here, we will introduce a method, known as *tableau representation* ⊞, that allows us to carry out the simplex method by hand in a much more convenient way.

In the simplex method, we have been maintaining and updating the matrix B^{-1} in each iteration. In the tableau representation, instead of dealing with B^{-1}, we maintain the matrix (or **tableau**) $B^{-1}[\boldsymbol{b}|A] = \begin{bmatrix}B^{-1}\boldsymbol{b} & B^{-1}\boldsymbol{A}_1 & \cdots & B^{-1}\boldsymbol{A}_n\end{bmatrix}$. Conventionally, the column $B^{-1}\boldsymbol{b}$ is called the **zeroth column** instead of the first column, and the column $B^{-1}\boldsymbol{A}_i$ is called the ith column (perhaps following the logic of zero-based indexing in coding </>?).

Remarks:

- *(Rationale for maintaining the matrix $B^{-1}[\boldsymbol{b}|A]$)* Note that left-multiplying B^{-1} on both sides of the equality constraints $\boldsymbol{b} = A\boldsymbol{x}$ gives $B^{-1}\boldsymbol{b} = B^{-1}A\boldsymbol{x}$. So the matrix $B^{-1}[\boldsymbol{b}|A]$ actually collects the coefficients of $B^{-1}\boldsymbol{b} = B^{-1}A\boldsymbol{x}$, which contain useful information for carrying out the simplex method.

- *(Zeroth column contains basic variables)* Recall from [1.2.12] that $A\boldsymbol{x}_B = B\boldsymbol{x}_B = \boldsymbol{b}$, so the vector of basic variables $\boldsymbol{x}_B$ is given by $B^{-1}\boldsymbol{b}$, which is precisely the zeroth column here.

2.2.8 **Performing some steps in the revised simplex method through the tableau.** After studying what the tableau is, we now consider how we should utilize it to perform the steps in the revised simplex method as suggested in [2.2.6]. For now, we focus on the steps (3)-(5); we will consider the rest in [2.2.9].

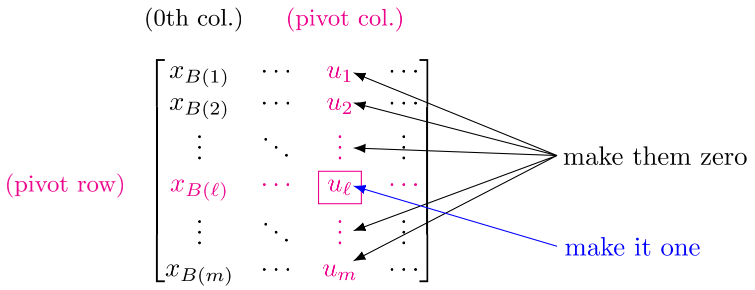
(3) Suppose that we are in the case where some entries of $\boldsymbol{u} = B^{-1}\boldsymbol{A}_j$ (the jth column of the tableau; also known as the **pivot column**) are positive. (Otherwise, the algorithm would have been terminated already.) Then we consider the tableau:

$$\begin{array}{cccc} \text{(0th col.)} & & \text{(pivot col.)} & \\ \left[\begin{matrix} x_{B(1)} \\ x_{B(2)} \\ \vdots \\ x_{B(m)} \end{matrix}\right. & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \begin{matrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{matrix} & \left.\begin{matrix} \cdots \\ \cdots \\ \vdots \\ \cdots \end{matrix}\right] \end{array}$$

For every row $i = 1, \ldots, m$, if $u_i > 0$, we compute the ratio $x_{B(i)}/u_i$. The index with the smallest ratio obtained is then the minimizing index ℓ, and such ratio $x_{B(\ell)}/u_\ell$ equals θ^*. That ℓth row is also known as the **pivot row**. The element that belongs to both the pivot column and the pivot row is called the **pivot element**, which is u_ℓ here.

(4-5) Here we would like to update the tableau $B^{-1}[\boldsymbol{b}|A]$ to $\bar{B}^{-1}[\boldsymbol{b}|A]$, where $\bar{B}$ being the new basis matrix as specified in step (4). This can be done by performing EROs on the tableau in the way suggested by the step (5) in [2.2.6], i.e., adding each of its rows a multiple of the *pivot row* to make all entries of the *pivot column* zero, except the *pivot element*, which is set to one.

We can also observe that, after performing such EROs, the zeroth column does contain the basic variables for the *new* basic feasible solution $\boldsymbol{x}^*$ (compare the EROs with the formulas in step (4)). Thus, performing these EROs allows us to do two things at once.



2.2.9 **Implementing the simplex method iterations through the tableau.** To implement the full simplex method, it is convenient to add one extra row (known as the **zeroth row**) at the top of the tableau, which contains the *negative* objective function value $-\boldsymbol{c}^T\boldsymbol{x} = -\boldsymbol{c}_B^T\boldsymbol{x}_B = -\boldsymbol{c}_B^T B^{-1}\boldsymbol{b}$ and the row vector of reduced costs $\bar{\boldsymbol{c}}^T = \boldsymbol{c}^T - \boldsymbol{c}_B^T B^{-1}A$:

$-\boldsymbol{c}_B^T B^{-1}\boldsymbol{b}$	$\boldsymbol{c}^T - \boldsymbol{c}_B^T B^{-1}A$
$B^{-1}\boldsymbol{b}$	$B^{-1}A$

It can be expressed more explicitly as:

$$\begin{array}{c|ccc}
-\boldsymbol{c}_B^T\boldsymbol{x}_B & \bar{c}_1 & \cdots & \bar{c}_n \\
\hline
x_{B(1)} & | & & | \\
\vdots & B^{-1}\boldsymbol{A}_1 & \cdots & B^{-1}\boldsymbol{A}_n \\
x_{B(m)} & | & & |
\end{array}$$

With the zeroth row added, we can perform the step (2) conveniently: just find out a column with negative reduced cost by inspection (assuming some reduced costs are negative), which corresponds to the non-basic index j. For the step (1), the initial choice of the basic feasible solution $\boldsymbol{x}$ is an *input* to this tableau method, and can be obtained as in [1.2.12].

It now remains to handle the update of the zeroth row in each iteration. After having the new basis matrix $\bar{B}$ from the previous iteration, we need to update the reduced costs and also the negative objective function value. As it turns out, the "rule" for updating the zeroth row works exactly the same as that for the other rows, namely adding a multiple of the pivot row to the zeroth row to make the entry at the pivot column (the "$\bar{c}_j$" originally) zero. This "rule" does not directly follow from the previous discussions on revised simplex method, and we will justify it in the following.

Proof. At the beginning of iteration, the zeroth row takes the form $[0|\boldsymbol{c}^T]-\boldsymbol{g}^T[\boldsymbol{b}|A]$ where $\boldsymbol{g}^T=\boldsymbol{c}_B^T B^{-1}$. Let column j and row ℓ be the pivot column and the pivot row respectively. Since the pivot row takes the form $\boldsymbol{h}^T[\boldsymbol{b}|A]$ (where $\boldsymbol{h}^T$ is the ℓth row of B^{-1}), adding a multiple of the pivot row to the zeroth row converts it to a row of the form $[0|\boldsymbol{c}^T]-\boldsymbol{p}^T[\boldsymbol{b}|A]$ for some vector $\boldsymbol{p}$.

Since the ERO we perform makes the pivot column entry of the zeroth row 0, we have $c_{\bar{B}(\ell)}-\boldsymbol{p}^T\boldsymbol{A}_{\bar{B}(\ell)}=c_j-\boldsymbol{p}^T\boldsymbol{A}_j=0$ (note that $j=\bar{B}(\ell)$). Now, consider the $\bar{B}(i)$th column with $i\neq\ell$, which corresponds to a original basic variable that stays in the new basis, so we know its zeroth row entry is 0 before the ERO. Also, for all $i\neq\ell$, we have $B^{-1}\boldsymbol{A}_{B(i)}=\boldsymbol{e}_i$ (as it is the ith column of the identity matrix $I_m=B^{-1}B$), meaning that the pivot row ($\ell\neq i$th row) entry of the ith column is also 0. Hence, after such ERO we know the zeroth row entry of the $\bar{B}(i)$th column is still 0. Therefore, the vector $\boldsymbol{p}$ satisfies $c_{\bar{B}(i)}-\boldsymbol{p}^T\boldsymbol{A}_{\bar{B}(i)}=0$ for all $i=1,\ldots,m$, which implies that $\boldsymbol{c}_{\bar{B}}^T-\boldsymbol{p}^T\bar{B}=\boldsymbol{0}$, or $\boldsymbol{p}^T=\boldsymbol{c}_{\bar{B}}^T\bar{B}^{-1}$. Consequently, the updated zeroth row after the ERO is given by

$$[0|\boldsymbol{c}^T]-\boldsymbol{c}_{\bar{B}}^T\bar{B}^{-1}[\boldsymbol{b}|A]=[-\boldsymbol{c}_{\bar{B}}^T\bar{B}^{-1}\boldsymbol{b}|\boldsymbol{c}^T-\boldsymbol{c}_{\bar{B}}^T\bar{B}^{-1}A]$$

which contains the correct quantities for the updated basis matrix $\bar{B}$. [Note: This explains why we consider *negative* objective function value rather than the objective function value itself.] □

2.2.10 **Procedure of the simplex method with tableau representation.**

(1) Start with the tableau associated with a basic feasible solution $\boldsymbol{x}$ and the corresponding basis matrix B (e.g., obtained as in [1.2.12]).

(2) Check the signs of the reduced costs $\bar{c}_1,\ldots,\bar{c}_n$ in the zeroth row of the tableau.

   i. If $\bar{c}_j\geq 0$ for all j, then the current basic feasible solution $\boldsymbol{x}$ is optimal, and the algorithm terminates ⏸.

   ii. Otherwise, choose a j with $\bar{c}_j<0$.

(3) Consider the pivot column (jth column) $\boldsymbol{u}=B^{-1}\boldsymbol{A}_j$ of the tableau. If no entry of $\boldsymbol{u}$ is positive, then the optimal value is $-\infty$, and the algorithm terminates ⏸.

(4) For each i where $u_i>0$, compute the ratio $x_{B(i)}/u_i$. Let ℓ be the index for a row that corresponds to the smallest ratio. Then the column $\boldsymbol{A}_{B(\ell)}$ exits the basis and the column $\boldsymbol{A}_j$ enters the basis.

(5) Add to each row of the tableau a multiple of the pivot row (ℓth row) such that the pivot element (u_ℓ originally) becomes one and all other entries of the pivot column become zero.

(6) Repeat (2)-(5) until the algorithm terminates ⏸.

In practice, when carrying out this procedure, the tableau would be filled with concrete values and it is helpful to add some extra labels to keep track of the variables corresponding to the rows/columns, like the following:

		x_1	x_2	x_3	x_4
	-3	$1/2$	-1	$5/3$	$1/3$
$x_1 =$	1	$1/3$	0	$-1/2$	$3/5$
$x_3 =$	3	$-5/2$	$\boxed{1}$	-3	$-1/2$
$x_4 =$	$7/6$	0	0	1	-2

Here the column $\boldsymbol{A}_3$ exits the basis and the column $\boldsymbol{A}_2$ enters the basis, and we would perform the ERO $\boldsymbol{r}_0 + \boldsymbol{r}_2 \to \boldsymbol{r}_0$:

		x_1	x_2	x_3	x_4
	0	-2	0	$-4/3$	$-1/6$
$x_1 =$	1	$1/3$	0	$-1/2$	$3/5$
$x_2 =$	3	$-5/2$	1	-3	$-1/2$
$x_4 =$	$7/6$	0	0	1	-2

Then we can repeat the procedure, say we choose $j = 3$ here:

		x_1	x_2	x_3	x_4
	0	-2	0	$-4/3$	$-1/6$
$x_1 =$	1	$1/3$	0	$-1/2$	$3/5$
$x_2 =$	3	$-5/2$	1	-3	$-1/2$
$x_4 =$	$7/6$	0	0	$\boxed{1}$	-2

2.3 Potential Issues in the Implementation

2.3.1 In Section 2.2, we have discussed how the simplex method can be implemented in practice and provided justifications on why it works. However, it turns out that there are indeed some potential issues when we are to implement the algorithms in practice, which are about *cycling* and *initialization*. In Section 2.3, we will investigate methods for handling these issues. Putting together the algorithms in Section 2.2 and the methods to be discussed in Section 2.3, we can develop a *two-phase algorithm*, which actually allows us to solve *any* standard form LP problem without issues.

2.3.2 **Cycling.** The first potential issue is known as *cycling* ♻. Cycling could arise when *degenerate* basic feasible solutions exist (assumed to be not the case in Section 2.2). It turns out that the simplex method we have developed could get stuck at an infinite loop over degenerate basic feasible solutions that are not optimal, and hence would never terminate. To deal with this issue, **pivoting rules** are developed, which are rules for selecting the columns/variables to exit/enter the basis, or the pivot column and the pivot row in the tableau.

2.3.3 **Bland's pivoting rule.** One pivoting rule that prevents cycling is known as the **Bland's pivoting rule**. The essential idea is to always choose the one with the *smallest index* when there are multiple possible candidates:

- *(Entering basis)* Find the smallest j where the reduced cost $\bar{c}_j$ is negative and let the column $\boldsymbol{A}_j$ enters the basis.
- *(Exiting basis)* Among all variables x_i that are tied in the test for choosing the exiting variable, select the one with the smallest value of i.

Example:

		x_1	x_2	x_3	x_4
	0	-2	0	$-4/3$	$-1/6$
$x_1 =$	1	$1/4$	0	$1/2$	$3/5$
$x_2 =$	2	$2/3$	1	-3	$-1/2$
$x_4 =$	$1/2$	$1/6$	0	1	-2

- *Pivot column: column 1.* Column 1 has the smallest index among all columns with negative reduced cost (columns 1, 3, and 4).
- *Pivot row: row 2.* Comparing the ratios $x_1/(1/4) = 4$, $x_2/(3/2) = 3$, and $x_4/(1/6) = 3$, the variables x_2 and x_4 achieve the smallest value, and x_2 is the one with the smallest index.

Despite the simplicity of the Bland's pivoting rule, it is indeed quite hard to *prove* that the rule can actually prevent cycling, so we will omit the proof here.

2.3.4 **Initialization.** Another potential issue arises from the very first step in the simplex method, namely choosing a basic feasible solution (with its associated basis matrix). While we have mentioned that a way to do that is to follow the procedure in [1.2.12], such way can actually lead to *wasted computational power* 🗑. This would occur if there is actually *no* basic feasible solution, which happens iff the feasible region is empty, as suggested by Corollary 1.2.e. In such case, following the procedure in [1.2.12] would just result in computing all possible basic solutions without getting any basic feasible solution (as there is none!).

To avoid such waste of computational power, here we will develop a (not so costly) method to check whether the feasible region is empty, to be applied before attempting to obtain an initial basic feasible solution. Such method also serves as an alternative to [1.2.12] for finding an initial basic feasible solution.

2.3.5 **Introducing an auxiliary LP problem.** The method to be introduced here utilizes an *auxiliary* LP problem, which helps us to determine whether the original LP problem is feasible, and also find a basic feasible solution if it is feasible.

Consider a standard form LP problem in general:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

By possibly multiplying some of the equality constraints by -1 (i.e., $\boldsymbol{a}_j^T\boldsymbol{x} = b_j \to -\boldsymbol{a}_j^T\boldsymbol{x} = -b_j$), we can assume WLOG that that $\boldsymbol{b} \geq \boldsymbol{0}$. Then, introduce a vector $\boldsymbol{y} \in \mathbb{R}^m$ of **artificial variables** and consider the following auxiliary LP problem:

$$\begin{aligned} \min \quad & y_1 + \cdots + y_m \\ \text{s.t.} \quad & A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0} \end{aligned}$$

This auxiliary problem can be solved by the simplex method without the issue of initialization, as we can always choose the initial basic feasible solution to be $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{0}, \boldsymbol{b})$, based on Theorem 1.2.c: Here the basic variables are $y_1, \ldots, y_m$, and since we can write the equality constraint as $A\boldsymbol{x} + \boldsymbol{y} = \begin{bmatrix} A & I_m \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \boldsymbol{b}$, they correspond to m linearly independent columns in the matrix $\begin{bmatrix} A & I_m \end{bmatrix}$, namely $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m$.

2.3.6 **Determining the feasibility of the original problem via the auxiliary problem.** The following result suggests how it can help us to determine the feasibility of the original standard form LP problem.

Proposition 2.3.a. The original standard form LP problem is feasible iff the optimal value of the auxiliary problem is 0.

Proof. "⇒": With the nonempty feasible region, we can pick a feasible solution $\boldsymbol{x}'$ to the original LP problem. Since the solution $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}', \boldsymbol{0})$ is a feasible solution for the auxiliary problem, and the objective function value of the auxiliary problem is always nonnegative on the feasible region, we conclude that the optimal value is 0.

"⇐": Assume that the optimal value of the auxiliary problem is 0. Then the corresponding optimal solution must be of the form $(\boldsymbol{x}, \boldsymbol{y}) = (\widetilde{\boldsymbol{x}}, \boldsymbol{0})$ where $A\widetilde{\boldsymbol{x}} + \boldsymbol{0} = A\widetilde{\boldsymbol{x}} = \boldsymbol{b}$ and $\widetilde{\boldsymbol{x}} \geq \boldsymbol{0}$. This means that $\widetilde{\boldsymbol{x}}$ is a feasible solution to the original LP problem. □

2.3.7 **Finding an initial basic feasible solution via the auxiliary problem.** As suggested by the proof of Proposition 2.3.a, if performing the simplex method on the auxiliary problem yields an optimal value of 0, then the resulting optimal solution would be of the form $(\boldsymbol{x}, \boldsymbol{y}) = (\widetilde{\boldsymbol{x}}, \mathbf{0})$, which is also a basic feasible solution of the feasible region in the *auxiliary* problem. The key idea is then to somehow "convert" this solution to a basic feasible solution of the feasible region in the *original* problem.

In the (very!) special case where the basic variables of $(x_1, \dots, x_n, y_1, \dots, y_m) = (\boldsymbol{x}, \boldsymbol{y}) = (\widetilde{\boldsymbol{x}}, \mathbf{0})$ are all contained in $\boldsymbol{x}$, say $x_{B(1)}, \dots, x_{B(m)}$, we immediately have that $\boldsymbol{x} = \widetilde{\boldsymbol{x}}$ is a basic feasible solution for the original problem. To see this, first note that the columns $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ would be linearly independent and $x_j = 0$ for all $j \notin \{B(1), \dots, B(m)\}$ by applying Theorem 1.2.c to the auxiliary problem. [Note: We have $A\boldsymbol{x} + \boldsymbol{y} = \begin{bmatrix} A & I_m \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}$.] Knowing these, we can apply Theorem 1.2.c again, but for the original problem this time, to deduce that $\boldsymbol{x} = \widetilde{\boldsymbol{x}}$ is a basic feasible solution for the original problem.

2.3.8 **Driving out artificial variables out of the basis.** However, generally speaking, it may very well be the case that some basic variables of $(\boldsymbol{x}, \boldsymbol{y}) = (\widetilde{\boldsymbol{x}}, \mathbf{0})$ are those "y"s (artificial variables). Let us denote the basic variables in this case by $x_{B(1)}, \dots, x_{B(k)}, y_{B(k+1)}, \dots, y_{B(m)}$, where $k < m$. Due to the presence of the artificial variables in the basic variables, we cannot immediately conclude that $\boldsymbol{x} = \widetilde{\boldsymbol{x}}$ is a basic feasible solution for the original problem.

Nevertheless, it is indeed possible to *drive all the artificial variables out of the basis* by performing suitable EROs on the tableau:

(1) For each $\ell = k + 1, \dots, m$,

  i. Pick a $j \in \{1, \dots, n\}$ such that the intersection of the ℓth row and the jth column in the tableau (namely the ℓth entry of $B^{-1}\boldsymbol{A}_j$) is nonzero. [Note: This is always possible since if there is no such j, then the ℓth entries of $B^{-1}\boldsymbol{A}_1, \dots, B^{-1}\boldsymbol{A}_n$ would all be zero, implying that A has linearly *dependent* rows, which has always been assumed to be not the case.]

  ii. Perform EROs on the tableau as in the step (5) of [2.2.10], with the pivot row and column being the ℓth row and the jth column respectively (so $y_{B(\ell)} = 0$ exits the basis and x_j enters the basis). [Note: It can be shown that after these EROs, we are actually still at the same basic feasible solution, but represented by a different basis.]

(2) The basic variables in the resulting basis would then consist of only the non-artificial variables "x"s, so we have driven all the artificial variables out of the basis ✓ (while the underlying basic feasible solution remains unchanged).

After this procedure has been done, we are again back to the special case above, and we know that $\boldsymbol{x} = \widetilde{\boldsymbol{x}}$ is a basic feasible solution for the original problem (associated with the basis obtained from the process above).

2.3.9 **Two-phase algorithm.** After discussing how we can deal with the potential issues arising from the simplex method, we can collect everything together to develop the following **two-phase algorithm** for solving any standard form LP, that terminates in finite amount of time (when a pivoting rule that prevents cycling is used, e.g., Bland's pivoting rule):

Phase I

(1) By multiplying some of the constraints in the problem by -1 if necessary, ensure that $\boldsymbol{b} \geq \mathbf{0}$.

(2) Solve the auxiliary problem by the simplex method.

  i. If the optimal value is positive, then the original LP is not feasible and the algorithm terminates ⏸.

  ii. If the optimal value is zero, then $\boldsymbol{x} = \widetilde{\boldsymbol{x}}$ is a basic feasible solution for the original problem, and its associated basis matrix is the final basis matrix for the auxiliary problem, obtained by driving all the artificial variables out of the basis (see [2.3.8]) if necessary.

Phase II

(1) Initialize the simplex method by using the basic feasible solution with the (final) basis matrix from Phase I.

(2) Update the zeroth row of the final tableau from Phase I based on the coefficients $\boldsymbol{c}^T$ in the original problem.

(3) Apply the simplex method on the updated tableau (see [2.2.10]).

2.3.10 **The big-M method.** The *big-M method* gives us a way to carry out the two phases in the two-phase algorithm *at once*, which relieves our computational burden when solving LP problems by hand. Therefore, the big-M method should serve as the main tool when you are solving LP problems by hand. The basic idea is to add a term in the original objective function that leads to *heavy penalization* when the newly added artificial variables y_i's are not all zero, so that the optimal solution would have y_i being all zero, hence feasible for the original LP problem (and also optimal by [2.3.10]b).

The big-M method is based on the following standard form LP problem (big-M problem) that is modified from the original:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} + M(y_1 + \cdots + y_m) \\ \text{s.t.} \quad & A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0} \end{aligned}$$

where $M > 0$ is a "large" constant (the $\underline{\text{big}}$ M). The following results provide the foundation for the big-M method:

(a) If the original LP problem is feasible and has a finite optimal value, then every optimal solution of the big-M problem satisfies $\boldsymbol{y} = \boldsymbol{0}$, provided that M is sufficiently large.

(b) If the big-M problem has an optimal solution of the form $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^*, \boldsymbol{0})$, then $\boldsymbol{x}^*$ is an optimal solution of the original LP problem.

(c) If the big-M problem has an optimal value of $-\infty$, then the original LP problem is either infeasible or has an optimal value of $-\infty$, provided that M is sufficiently large.

Proof.

(a) Assume to the contrary that the original LP problem is feasible and has a finite optimal value, while there is an optimal solution $(\widetilde{\boldsymbol{x}}, \boldsymbol{y})$ of the big-M problem satisfying $y_k > 0$ for some $k = 1, \dots, m$. Let $\boldsymbol{x}^*$ denote an optimal solution of the original LP problem, which exists due to the finiteness of the optimal value. Then, note that $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^*, \boldsymbol{0})$ is a feasible solution of the big-M problem. With M being sufficiently large, we would have

$$\boldsymbol{c}^T\boldsymbol{x}^* + M\boldsymbol{0} < \boldsymbol{c}^T\widetilde{\boldsymbol{x}} + My_k \leq \boldsymbol{c}^T\widetilde{\boldsymbol{x}} + M\sum_{i=1}^{m} y_i,$$

which contradicts to the optimality of $(\widetilde{\boldsymbol{x}}, \boldsymbol{y})$.

(b) Since $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^*, \boldsymbol{0})$ is optimal for the big-M problem, we have $\boldsymbol{c}^T\boldsymbol{x}^* \leq \boldsymbol{c}^T\boldsymbol{x} + M(y_1 + \cdots + y_m)$ for every feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ of the big-M problem. Now fix any feasible solution $\boldsymbol{x}$ of the original LP problem, which satisfies $A\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$. Hence, $(\boldsymbol{x}, \boldsymbol{0})$ is a feasible solution of the big-M problem, which implies that $\boldsymbol{c}^T\boldsymbol{x}^* \leq \boldsymbol{c}^T\boldsymbol{x} + M(0) = \boldsymbol{c}^T\boldsymbol{x}$.

(c) Assume to the contrary that the big-M problem has an optimal value of $-\infty$, while the original LP problem has an optimal solution $\boldsymbol{x}^*$. Then, after applying the simplex method on the problem, at the termination we would get a nonzero feasible direction $\boldsymbol{d} = (\boldsymbol{d}_{\boldsymbol{x}}, \boldsymbol{d}_{\boldsymbol{y}}) \geq \boldsymbol{0}$ at a feasible solution $(\boldsymbol{x}, \boldsymbol{y})$, that leads a decrease in the objective function value, i.e., $\boldsymbol{c}^T\boldsymbol{d}_{\boldsymbol{x}} + M\sum_{i=1}^{m} d_{y_i} < 0$.
With M being sufficiently large, we must have $d_{y_i} = 0$ for all $i = 1, \dots, m$; otherwise the inequality above would be contradicted by setting sufficiently large M. Combining this with the inequality above gives $\boldsymbol{c}^T\boldsymbol{d}_{\boldsymbol{x}} < 0$.
Since $\boldsymbol{d} \geq \boldsymbol{0}$ is a feasible direction at $(\boldsymbol{x}, \boldsymbol{y})$ for the big-M problem, we have $A(\boldsymbol{x}+\boldsymbol{d}_{\boldsymbol{x}})+(\boldsymbol{y}+\boldsymbol{d}_{\boldsymbol{y}}) = \boldsymbol{0}$, which implies $A\boldsymbol{d}_{\boldsymbol{x}} = \boldsymbol{b} - (A\boldsymbol{x} + \boldsymbol{y}) = \boldsymbol{b} - \boldsymbol{b} = \boldsymbol{0}$. Hence, $\boldsymbol{d}_{\boldsymbol{x}}$ is a feasible direction at $\boldsymbol{x}^*$ also. But then $\boldsymbol{c}^T\boldsymbol{d}_{\boldsymbol{x}} < 0$ would imply that $\boldsymbol{c}^T(\boldsymbol{x}^* + \boldsymbol{d}_{\boldsymbol{x}}) < \boldsymbol{c}^T\boldsymbol{x}^*$, contradicting to the optimality of $\boldsymbol{x}^*$.

□

The procedure for the **big-M method** is thus as follows.

(1) Solve the big-M problem by the simplex method, with the initial basic feasible solution being $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{0}, \boldsymbol{b})$ (see [2.2.10]).

[Note: In practice, to ensure the applicability of [2.3.10]a, we often leave M as an undetermined large constant during the simplex iterations (e.g., in the tableau method): We always have $M > c$ when comparing M with any other constant c and we do not specify the exact value of M.]

(2) i. If the optimal solution obtained is of the form $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^*, \boldsymbol{0})$, then $\boldsymbol{x}^*$ is an optimal solution of the original LP problem (by [2.3.10]b).

ii. If the optimal solution obtained does not satisfy $\boldsymbol{y} = \boldsymbol{0}$ or the big-M problem has an optimal value of $-\infty$, then the original LP problem is either infeasible or has an optimal value of $-\infty$. [Note: Here we use [2.3.10]c and the contrapositive of [2.3.10]a.]

3 Duality Theory

3.0.1 In our context here, *duality* refers to the connection between two LP problems, which are called *primal* and *dual* problems respectively. The LP problems we have been working with so far can all be treated as *primal* problems. Duality theory provides us another point of view on those LP problems through their duals, thereby giving us: (i) new tools to analyze those LP problems by *weak duality* and *strong duality* (Theorems 3.2.a and 3.2.d), (ii) new optimality conditions like *complementary slackness* ([3.2.5]), and (iii) new algorithms for solving LPs like *dual simplex method* (Section 3.3). Indeed, for general optimization problems, duality plays an important role also.

3.1 Primal and Dual Problems

3.1.1 **Definitions of primal and dual problems.** Let A be a matrix with rows $\boldsymbol{a}_1^T, \dots, \boldsymbol{a}_m^T$ and columns $\boldsymbol{A}_1, \dots, \boldsymbol{A}_n$. A **primal** problem refers to a general LP problem of minimization here:

$$\begin{aligned}
\min \quad & \boldsymbol{c}^T\boldsymbol{x} \\
\text{s.t.} \quad & \boldsymbol{a}_j^T\boldsymbol{x} \geq b_j \text{ for all } j \in M_1, \\
& \boldsymbol{a}_j^T\boldsymbol{x} \leq b_j \text{ for all } j \in M_2, \\
& \boldsymbol{a}_j^T\boldsymbol{x} = b_j \text{ for all } j \in M_3, \\
& x_i \geq 0 \text{ for all } i \in N_1, \\
& x_i \leq 0 \text{ for all } i \in N_2, \\
& x_i \text{ free for all } i \in N_3.
\end{aligned}$$

Its **dual** is then defined as the following LP problem of maximization:

$$\begin{aligned}
\max \quad & \boldsymbol{b}^T\boldsymbol{y} \\
\text{s.t.} \quad & y_j \geq 0 \text{ for all } j \in M_1, \\
& y_j \leq 0 \text{ for all } j \in M_2, \\
& y_j \text{ free for all } j \in M_3, \\
& \boldsymbol{A}_i^T\boldsymbol{y} \leq c_i \text{ for all } i \in N_1, \\
& \boldsymbol{A}_i^T\boldsymbol{y} \geq c_i \text{ for all } i \in N_2, \\
& \boldsymbol{A}_i^T\boldsymbol{y} = c_i \text{ for all } i \in N_3.
\end{aligned}$$

The following table tells us the "rule" for "translating" between primal and dual problems:

Primal	minimize	maximize	Dual
constraints	$\geq b_j$ $\leq b_j$ $= b_j$	≥ 0 ≤ 0 free	variables
variables	≥ 0 ≤ 0 free	$\leq c_i$ $\geq c_i$ $= c_i$	constraints

3.1.2 **The dual of the dual is the primal.** One nice property of the dual can be summarized in one sentence: *The dual of the dual is the primal.* This means that taking dual twice would give you back the original LP problem. To see why this holds, consider the following:

(1) Start with a primal:

$$\begin{aligned} \min\quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.}\quad & \boldsymbol{a}_j^T\boldsymbol{x} \geq b_j \text{ for all } j \in M_1, \\ & \boldsymbol{a}_j^T\boldsymbol{x} \leq b_j \text{ for all } j \in M_2, \\ & \boldsymbol{a}_j^T\boldsymbol{x} = b_j \text{ for all } j \in M_3, \\ & x_i \geq 0 \text{ for all } i \in N_1, \\ & x_i \leq 0 \text{ for all } i \in N_2, \\ & x_i \text{ free for all } i \in N_3. \end{aligned}$$

(2) Its dual is:

$$\begin{aligned} \max\quad & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.}\quad & y_j \geq 0 \text{ for all } j \in M_1, \\ & y_j \leq 0 \text{ for all } j \in M_2, \\ & y_j \text{ free for all } j \in M_3, \\ & \boldsymbol{A}_i^T\boldsymbol{y} \leq c_i \text{ for all } i \in N_1, \\ & \boldsymbol{A}_i^T\boldsymbol{y} \geq c_i \text{ for all } i \in N_2, \\ & \boldsymbol{A}_i^T\boldsymbol{y} = c_i \text{ for all } i \in N_3. \end{aligned}$$

(3) To get the "dual of the dual", we need to first convert the dual problem above to a minimization problem as follows:

$$\begin{aligned} \min\quad & (-\boldsymbol{b})^T\boldsymbol{y} \\ \text{s.t.}\quad & (-\boldsymbol{A}_i)^T\boldsymbol{y} \geq -c_i \text{ for all } i \in N_1, \\ & (-\boldsymbol{A}_i)^T\boldsymbol{y} \leq -c_i \text{ for all } i \in N_2, \\ & (-\boldsymbol{A}_i)^T\boldsymbol{y} = -c_i \text{ for all } i \in N_3, \\ & y_j \geq 0 \text{ for all } j \in M_1, \\ & y_j \leq 0 \text{ for all } j \in M_2, \\ & y_j \text{ free for all } j \in M_3. \end{aligned}$$

(4) Viewing $-\boldsymbol{A}_i$'s as the rows of $-A^T$ and $-\boldsymbol{a}_j$'s as the columns of $-A^T$, the dual of this problem is then:

$$\begin{aligned} \max\quad & (-\boldsymbol{c})^T\boldsymbol{x} \\ \text{s.t.}\quad & x_i \geq 0 \text{ for all } i \in N_1, \\ & x_i \leq 0 \text{ for all } i \in N_2, \\ & x_i \text{ free for all } i \in N_3, \\ & (-\boldsymbol{a}_j)^T\boldsymbol{x} \leq -b_j \text{ for all } j \in M_1, \\ & (-\boldsymbol{a}_j)^T\boldsymbol{x} \geq -b_j \text{ for all } j \in M_2, \\ & (-\boldsymbol{a}_j)^T\boldsymbol{x} = -b_j \text{ for all } j \in M_3. \end{aligned}$$

(5) Converting it back to a minimization problem yields the original primal:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{a}_j^T\boldsymbol{x} \geq b_j \text{ for all } j \in M_1, \\ & \boldsymbol{a}_j^T\boldsymbol{x} \leq b_j \text{ for all } j \in M_2, \\ & \boldsymbol{a}_j^T\boldsymbol{x} = b_j \text{ for all } j \in M_3, \\ & x_i \geq 0 \text{ for all } i \in N_1, \\ & x_i \leq 0 \text{ for all } i \in N_2, \\ & x_i \text{ free for all } i \in N_3. \end{aligned}$$

3.1.3 **Duality for standard form LP problems.** For standard form LP problems, the expressions of primal and dual can be significantly simplified as follows:

$$\begin{array}{llcll} & \underline{\text{Primal}} & \qquad\qquad & & \underline{\text{Dual}} \\ \min & \boldsymbol{c}^T\boldsymbol{x} & & \max & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, & & \text{s.t.} & A^T\boldsymbol{y} \leq \boldsymbol{c}. \\ & \boldsymbol{x} \geq \boldsymbol{0}. & & & \end{array}$$

By reducing a general LP problem to an equivalent standard form problem (through the operations mentioned in [1.1.7]), we can study the duality for the general LP problem through the simpler duality for standard form LP problem, as the following result suggests:

Theorem 3.1.a. Suppose that we have transformed a general LP problem Π_1 to an equivalent standard form LP problem Π_2 (through the operations in [1.1.7]). Let D_1 and D_2 be the duals of Π_1 and Π_2 respectively. Then, D_1 and D_2 are either both infeasible or have the same finite optimal value.

Proof. Omitted. □

Based on Theorem 3.1.a, we can have the following route to study the duality of general LP problem Π_1 via the duality for standard form LP problem.

$$\begin{tikzcd} \Pi_1 \arrow[d, "\text{transform by [1.1.7]}"'] & & D_1 \\ \Pi_2 \arrow[rr, "\text{duality for standard form LP problem}"'] & & D_2 \arrow[u, "\text{Theorem 3.1.a}"'] \end{tikzcd}$$

3.2 Duality Theorems

3.2.1 Two major duality theorems that relate the primal and dual problems are *weak duality* and *strong duality*, which establish *inequality* and *equality* on the optimal values of the primal and dual problems respectively.

3.2.2 **Weak duality.** As the following result suggests, the optimal value of the dual is always less than or equal to the optimal value of the primal; indeed the same is true for general *objective function value*:

Theorem 3.2.a (Weak duality)**.** If $\boldsymbol{x}$ and $\boldsymbol{y}$ are feasible solutions to the primal and the dual problems respectively, then $\boldsymbol{b}^T\boldsymbol{y} \leq \boldsymbol{c}^T\boldsymbol{x}$.

Proof. Fix any feasible solution $\boldsymbol{x}$ to the primal and feasible solution $\boldsymbol{y}$ to the dual. Let $u_i := y_i(\boldsymbol{a}_i^T\boldsymbol{x} - b_i)$ and $v_j := (c_j - \boldsymbol{A}_j^T\boldsymbol{y})x_j$. By the primal and dual feasibility, the signs of y_i and $\boldsymbol{a}_i^T\boldsymbol{x} - b_i$ are the same, so do the signs of $c_j - \boldsymbol{A}_j^T\boldsymbol{y}$ and x_j. Hence, we must have $u_i \geq 0$ for every i and $v_j \geq 0$ for every j. This then implies that

$$\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y} = (\boldsymbol{y}^T A\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y}) - (\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{x}^T A^T\boldsymbol{y}) = \sum_i u_i + \sum_j v_j \geq 0,$$

as desired. □

Based on weak duality, we can obtain the following tests for the feasibility of the primal and dual problems.

Corollary 3.2.b.

(a) If the optimal value of the primal is $-\infty$, then the dual problem must be infeasible.

(b) If the optimal value of the dual is ∞, then the primal problem must be infeasible.

Proof.

(a) Assume to the contrary the optimal value of the primal is $-\infty$ while there exists a feasible solution $\boldsymbol{y}$ to the dual problem. Then, by weak duality, we have $\boldsymbol{b}^T\boldsymbol{y} \leq \boldsymbol{c}^T\boldsymbol{x}$ for every feasible solution $\boldsymbol{x}$ to the primal problem, which would then imply that the objective function for the primal problem is bounded below (by $\boldsymbol{b}^T\boldsymbol{y} \in \mathbb{R}$) on the feasible region, thus the optimal value of the primal would not be $-\infty$, contradiction.

(b) Similar to (a).

□

Another helpful result as a corollary of the weak duality provides an optimality condition for both primal and dual. Later in [3.2.5], we will have more discussions on the optimality condition.

Corollary 3.2.c. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be feasible solutions to the primal and the dual problems respectively. If $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{b}^T\boldsymbol{y}$, then $\boldsymbol{x}$ and $\boldsymbol{y}$ are optimal solutions to the primal and the dual respectively.

Proof. We first show the optimality of $\boldsymbol{x}$ for the primal. Fix any feasible solution $\boldsymbol{z}$ to the primal. Then, we have $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{b}^T\boldsymbol{y} \overset{\text{(weak duality)}}{\leq} \boldsymbol{c}^T\boldsymbol{z}$, establishing the optimality of $\boldsymbol{x}$.

Next, we show the optimality of $\boldsymbol{y}$ for the dual. Fix any feasible solution $\boldsymbol{w}$ to the dual. Then, we have $\boldsymbol{b}^T\boldsymbol{y} = \boldsymbol{c}^T\boldsymbol{x} \overset{\text{(weak duality)}}{\geq} \boldsymbol{b}^T\boldsymbol{w}$, establishing the optimality of $\boldsymbol{y}$. □

3.2.3 **Strong duality.** The next important duality theorem is the *strong duality*, which is often considered to be the *central result* for studying duality of LP problems.

Theorem 3.2.d (Strong duality)**.** If a LP problem has an optimal solution, then so does its dual, and the respective optimal values are equal.

Proof. We first prove the result for the case with standard form primal.

$$\begin{array}{llcll} & \underline{\text{Primal}} & \qquad & & \underline{\text{Dual}} \\ \min & \boldsymbol{c}^T\boldsymbol{x} & & \max & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, & & \text{s.t.} & A^T\boldsymbol{y} \leq \boldsymbol{c}. \\ & \boldsymbol{x} \geq \boldsymbol{0}. & & & \end{array}$$

Applying the simplex method on the primal problem, at the termination we would get an optimal basic feasible solution $\boldsymbol{x}$ with an optimal basis matrix B. Let $\boldsymbol{x}_B := B^{-1}\boldsymbol{b}$ denote the vector of basic variables. Also, we know the reduced costs must all be nonnegative at the termination, so $\bar{\boldsymbol{c}} = \boldsymbol{c} - A^T(B^{-1})^T\boldsymbol{c}_B \geq \boldsymbol{0}$.

Now we choose $\boldsymbol{y} = (B^{-1})^T\boldsymbol{c}_B$. Since $\bar{\boldsymbol{c}} \geq \boldsymbol{0}$, we have $A^T\boldsymbol{y} \leq \boldsymbol{c}$, hence $\boldsymbol{y}$ is a feasible solution to the dual problem. Also, we have $\boldsymbol{b}^T\boldsymbol{y} = \boldsymbol{b}^T(B^{-1})^T\boldsymbol{c}_B = \boldsymbol{c}_B^T B^{-1}\boldsymbol{b} = \boldsymbol{c}_B\boldsymbol{x}_B = \boldsymbol{c}^T\boldsymbol{x}$. Thus, by Corollary 3.2.c, $\boldsymbol{y}$ serves as an optimal solution to the dual problem, and also the respective optimal values are equal as $\boldsymbol{b}^T\boldsymbol{y} = \boldsymbol{c}^T\boldsymbol{x}$.

Now, for the general case where the primal Π_1 is a general LP problem, we first transform it to an equivalent standard form LP problem Π_2 (through [1.1.7]). Now let D_1 and D_2 be the duals of Π_1 and Π_2 respectively. From the special case proven above, Π_2 and D_2 have the same optimal value.

From this we deduce that D_2 must be feasible, so by Theorem 3.1.a, D_1 and D_2 would have the same optimal value. Also, from the equivalence, Π_1 and Π_2 have the same optimal value by Proposition 1.1.a. This shows that Π_1 and D_1 have the same optimal value. □

3.2.4 **Possibilities for the primal and the dual.** By Corollary 1.2.g, for every LP problem, it falls into exactly one of the following three categories:

(1) It has an optimal solution, and the optimal value is finite.

(2) The optimal value is $-\infty$ (for minimization) or ∞ (for maximization).

(3) The problem is infeasible.

By weak duality (more specifically, Corollary 3.2.b) and strong duality, we can deduce all possible combinations of the categories for the primal and dual problems:

Primal Dual	Optimal solution exists	Optimal value is $-\infty$	Infeasible
Optimal solution exists	✓ (equal optimal value)	✗	✗
Optimal value is ∞	✗	✗	✓
Infeasible	✗	✓	✓

[Note: Technically the results do not assure that it is possible for both primal and dual problems to be infeasible. But it is indeed not hard to construct an example where this is the case, e.g.:

$$\begin{array}{ll} & \underline{\text{Primal}} \\ \text{min} & x_1 + 2x_2 \\ \text{s.t.} & x_1 + x_2 = 1, \\ & 2x_1 + 2x_2 = 3. \end{array} \qquad \begin{array}{ll} & \underline{\text{Dual}} \\ \text{max} & y_1 + 3y_2 \\ \text{s.t.} & y_1 + 2y_2 = 1, \\ & y_1 + 2y_2 = 2. \end{array}$$

]

3.2.5 **Complementary slackness.** *Complementary slackness* provides a necessary and sufficient condition for $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ to be optimal for the primal and dual problems respectively. For simplicity, here we shall only focus on the case with standard form primal:

Theorem 3.2.e (Optimality condition based on complementary slackness)**.** Consider a primal problem and its dual:

$$\begin{array}{ll} & \underline{\text{Primal}} \\ \min & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{array} \qquad \begin{array}{ll} & \underline{\text{Dual}} \\ \max & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} & A^T\boldsymbol{y} \leq \boldsymbol{c}. \end{array}$$

The vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ are optimal solutions to the primal and the dual problems respectively iff

(1) *(Primal feasibility)* $A\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$.

(2) *(Dual feasibility)* $A^T\boldsymbol{y} \leq \boldsymbol{c}$.

(3) *(Complementary slackness)* $(c_i - \boldsymbol{A}_i^T\boldsymbol{y})x_i = 0$ for every $i = 1, \dots, n$, where $\boldsymbol{A}_i$ is the ith column of A for every i.

Proof. "⇒": Suppose that $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ are optimal solutions, hence feasible solutions, to the primal and the dual respectively. Then the primal feasibility and dual feasibility are automatically satisfied, so it remains to show the complementary slackness.

By strong duality, we have $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{b}^T\boldsymbol{y}$, which implies that

$$\sum_{i=1}^{n}(c_i - \boldsymbol{A}_i^T\boldsymbol{y})x_i = (\boldsymbol{c} - A^T\boldsymbol{y})^T\boldsymbol{x} = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{x}^TA^T\boldsymbol{y} = \boldsymbol{c}^T\boldsymbol{x} - (A\boldsymbol{x})^T\boldsymbol{y} = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y} = 0.$$

Since both $\boldsymbol{x}$ and $\boldsymbol{y}$ are feasible, we have $\boldsymbol{A}_i^T\boldsymbol{y} \leq c_i$ and $x_i \geq 0$ for every $i = 1, \dots, n$, which suggests that each summand is nonnegative. This then forces that $(c_i - \boldsymbol{A}_i^T\boldsymbol{y})x_i = 0$ for every $i = 1, \dots, n$.

"⇐": Suppose that the three conditions hold. Then, like above, we have $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y} = \sum_{i=1}^{n}(c_i - \boldsymbol{A}_i^T\boldsymbol{y})x_i = 0$, thus $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{b}^T\boldsymbol{y}$. Hence, by Corollary 3.2.c, $\boldsymbol{x}$ and $\boldsymbol{y}$ are optimal solutions to the primal and the dual respectively. □

[Note: The condition that $(c_i - \boldsymbol{A}_i^T\boldsymbol{y})x_i = 0$ for every $i = 1, \dots, n$ is called *complementary slackness* because it asserts that for every $i = 1, \dots, n$, one of the inequalities $x_i \geq 0$ (primal) and $\boldsymbol{A}_i^T\boldsymbol{y} \leq c_i$ (dual) must hold *tightly* without "slack" (i.e., the equality holds); otherwise the expression would be greater than 0. In other words, the appearance of "slackness" in one inequality must be complemented by the "non-slackness" in another.]

3.3 Dual Simplex Method

3.3.1 The *dual simplex method* is an alternative algorithm to the simplex method investigated in Section 2. Here, we will study how it works, and how the tableau introduced before helps us carry out the dual simplex method. Like the simplex method, here we will also focus on standard form LP problems only.

3.3.2 **Motivation.** To motivate the dual simplex method, we consider the following lemma.

Lemma 3.3.a. Consider a primal problem and its dual:

$$\begin{array}{llcll} & \underline{\text{Primal}} & & & \underline{\text{Dual}} \\ \min & \boldsymbol{c}^T\boldsymbol{x} & & \max & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, & & \text{s.t.} & A^T\boldsymbol{y} \leq \boldsymbol{c}. \\ & \boldsymbol{x} \geq \boldsymbol{0}. & & & \end{array}$$

Let $B \in \mathbb{R}^{m\times m}$ be a basis matrix for the primal problem, corresponding to a basic feasible solution $\boldsymbol{x}$, with $\boldsymbol{x}_B := B^{-1}\boldsymbol{b} \in \mathbb{R}^n$ being the vector of basic variables. Let $\boldsymbol{y}_B := (B^{-1})^T\boldsymbol{c}_B \in \mathbb{R}^m$. Then:

(a) The vector $\boldsymbol{y}_B$ is a basic solution to the dual problem.

(b) The basis matrix B is optimal if $\boldsymbol{x}_B \geq \boldsymbol{0}$ *(primal feasibility)* and $A^T\boldsymbol{y}_B \leq \boldsymbol{c}$ *(dual feasibility)*.

Proof.

(a) From the basis matrix B, we know that there are m basic indices $B(1), \dots, B(m)$ corresponding to the basic variables. By [2.1.5], the reduced cost is $\bar{c}_{B(i)} = c_{B(i)} - \boldsymbol{c}_B^TB^{-1}\boldsymbol{A}_{B(i)} = 0$ for every $i = 1, \dots, m$, where $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent. This implies that $\boldsymbol{A}_{B(i)}^T\boldsymbol{y}_B = \boldsymbol{A}_{B(i)}^T(B^{-1})^T\boldsymbol{c}_B = \boldsymbol{c}_B^TB^{-1}\boldsymbol{A}_{B(i)} = c_{B(i)}$ for every $i = 1, \dots, m$, which gives us m linearly independent active constraints at $\boldsymbol{y}_B$ (for the dual problem). Thus $\boldsymbol{y}_B$ is a basic solution to the dual problem.

(b) The condition $\boldsymbol{x}_B \geq \mathbf{0}$ is precisely the feasibility requirement for B to be optimal, so it suffices to establish the optimality requirement, namely that all the reduced costs are nonnegative (for the primal problem). Using a similar argument as in (a), we can express the reduced costs as $\bar{\boldsymbol{c}} = \boldsymbol{c} - A^T\boldsymbol{y}_B$. Therefore, the condition $A^T\boldsymbol{y}_B \leq \boldsymbol{c}$ is indeed equivalent to $\bar{\boldsymbol{c}} \geq \mathbf{0}$, showing that the optimality requirement is satisfied.

□

Remarks:

- For (b), one can verify that the *complementary slackness* property is also satisfied by $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y}_B \in \mathbb{R}^m$ under such conditions: For every $j = 1, \ldots, n-m$, we have $x_{N(j)} = 0$; for every $i = 1, \ldots, m$, we have $c_{B(i)} - \boldsymbol{A}_{B(i)}^T\boldsymbol{y}_B \overset{\text{(proof above)}}{=} 0$. Hence, by Theorem 3.2.e, we can conclude that $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y}_B \in \mathbb{R}^m$ are optimal solutions to the primal and the dual problems respectively, under such conditions.
- The vector $\boldsymbol{y}_B = (B^{-1})^T\boldsymbol{c}_B$ is sometimes known as the **dual basic solution** associated with the basis matrix B.

3.3.3 **Comparing simplex method and dual simplex method.** From the proof of Lemma 3.3.a, we see that the nonnegativity of reduced costs in the primal problem actually corresponds to the dual feasibility for a basic solution $\boldsymbol{y}_B$ to the dual problem. Hence, we can view the (primal) simplex method as an algorithm that keeps changing the basis matrix B such that the primal feasibility (for the basic solution with vector of basic variables being $\boldsymbol{x}_B$) is maintained, and terminates if the dual feasibility (for the basic solution $\boldsymbol{y}_B$) is achieved.

On the other hand, the dual simplex method can be viewed as an algorithm that keeps changing the basis matrix B such that the dual feasibility is maintained, and terminates if the primal feasibility is achieved. This gives us the basic idea of how the dual simplex method works; let us study how it can actually be carried based on the simplex tableau introduced before next.

3.3.4 **Procedure for the dual simplex method.**

(1) Start with the simplex tableau associated with a basis matrix such that all reduced costs are nonnegative *(dual feasibility)*.

(2) Check the signs of the basic variables $x_{B(1)}, \ldots, x_{B(m)}$ in the zeroth column of the tableau.

   i. If $x_{B(i)} \geq 0$ for all i, then the current basic feasible solution $\boldsymbol{x}$ is optimal (for the primal problem), and the algorithm terminates ⏸.

   ii. Otherwise, choose a ℓ such that $x_{B(\ell)} < 0$.

(3) Consider the ℓth row (the **pivot row**) with entries $x_{B(\ell)}, v_1, \ldots, v_n$. If no entry of the row is negative, then the optimal value for the dual problem is ∞, which implies the infeasibility of the primal problem, and then the algorithm terminates.

(4) For every i where $v_i < 0$, compute the ratio $\bar{c}_i/|v_i|$. Let j be the index for a column that corresponds to the smallest ratio (the **pivot column**). Then the column $\boldsymbol{A}_{B(\ell)}$ exits the basis and the column $\boldsymbol{A}_j$ enters the basis.

(5) Add to each row of the tableau a multiple of the pivot row (ℓth row) such that v_j (the **pivot element**, which is the intersection between the pivot row and the pivot column) becomes 1 and all other entries of the pivot column become 0.

(6) Repeat (2)-(5) until the algorithm terminates ⏸.

Remarks:

- *(Similarity to the simplex method)* As one can see, this procedure is quite analogous to the one for the simplex method (see [2.2.10]), just with something "flipped".

- *(Usefulness)* Since the dual simplex method often turns out to work better than the simplex method in practice, it is considered to be practically useful. Besides, the dual simplex method is usually helpful to deal with the case where the vector $\boldsymbol{b}$ in the primal problem changes, since such changes would affect only the zeroth column of the final tableau obtained by the simplex method, possibly making some entries there negative. In such case, one cannot directly carry out the (primal) simplex method on that tableau, but the dual simplex method still works well.

4 Sensitivity Analysis

4.0.1 **Why sensitivity analysis?** Consider a standard form LP problem:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

Previously, we have focused on developing methods for solving it and obtaining optimal solutions, with the parameters A, $\boldsymbol{b}$, and $\boldsymbol{c}$ being fixed. However, in practice, there are often changes to A, $\boldsymbol{b}$, and $\boldsymbol{c}$ from the updated information (e.g., some raw materials 🧀 become more expensive and some goods 🍔 can be sold at higher prices), which makes the previously obtained optimal solution not necessarily optimal for the new LP problem anymore.

While one can certainly solve the new LP problem from scratch again to get a new optimal solution, this approach may not be efficient and acceptable (e.g., it would not be acceptable if it takes a day to solve for the new optimal solution from scratch, while updates occur daily). Therefore, in Section 4, we will develop some ways to *efficiently* analyze/estimate the impacts from changing the LP problems in certain aspects, without solving the new LP problem from scratch. This is known as *sensitivity analysis*.

For mathematical tractability, throughout we shall focus on standard form LP problems, with the usual assumption that the rows of A are linearly independent.

4.1 Local Sensitivity Analysis

4.1.1 **Basic idea.** The first type of sensitivity analysis to be discussed is *local sensitivity analysis*. As suggested by the term "local", it refers to the case where the changes are "small". The intuitive idea of local sensitivity analysis is that the previous obtained optimal solution should remain optimal after "sufficiently small" changes.

Consider a standard form LP:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

Suppose that we have obtained an optimal solution $\boldsymbol{x}^*$ to this LP with an optimal basis matrix B. By the definition of optimal basis matrix, we have:

- *(feasibility)* $B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$, and
- *(optimality)* $\bar{\boldsymbol{c}}^T = \boldsymbol{c}^T - \boldsymbol{c}_B^T B^{-1} A \geq \boldsymbol{0}$ ($\boldsymbol{0}$ is a row vector), or $\bar{\boldsymbol{c}} = \boldsymbol{c} - A^T (B^{-1})^T \boldsymbol{c}_B \geq \boldsymbol{0}$ ($\boldsymbol{0}$ is a column vector) after taking transpose.

After having some changes, we investigate whether these two conditions are affected. If both of them remain to hold, then the original basis matrix B would remain optimal, preserving the optimality. Based on this investigation, we can then deduce how "small" the changes should be to ensure the preservation of optimality. To illustrate how this approach works, we will go through several case studies on the possible changes to the LP problem in the following.

4.1.2 **Case Study 1: Changes in the vector $\boldsymbol{c}$.** Here we consider the special case where we change $c_i \to c_i + \delta$ for a certain $i = 1, \ldots, n$ only. Because there is no change on A and $\boldsymbol{b}$, the feasibility condition $B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$ always remains valid. So, for the basis matrix B to remain optimal, we just need $\bar{\boldsymbol{c}}' = \boldsymbol{c}' - A^T (B^{-1})^T \boldsymbol{c}'_B \geq \boldsymbol{0}$. The situation can be divided into two cases, and we will analyze (i) the condition for preserving optimality and (ii) change in optimal value (assuming the condition for preserving optimality is satisfied):

- *Case 1: x_i is a nonbasic variable.* In this case, changing $c_i \to c_i + \delta$ would not affect $\boldsymbol{c}_B$, so $\boldsymbol{c}'_B = \boldsymbol{c}_B$. Hence, $\bar{c}'_j \geq 0$ for every $j \neq i$.
  (1) *Condition for preserving optimality:* To ensure that $\bar{c}'_i \geq 0$ also, we would need

$$\bar{c}'_i \geq 0 \iff \underbrace{c_i - \boldsymbol{A}_i^T (B^{-1})^T \boldsymbol{c}_B}_{\bar{c}_i} + \delta \geq 0 \iff \boxed{\delta \geq -\bar{c}_i}.$$

  (2) *Change in optimal value:* Since we can write $\boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{c}_B^T \boldsymbol{x}_B$ and $\boldsymbol{c}_B$ is not affected by the change, the change in optimal value is $\boxed{0}$.

- *Case 2: x_i is a basic variable.* This case is more complicated as changing $c_i \to c_i + \delta$ would affect $\boldsymbol{c}_B$. Writing $i = B(\ell)$ for some ℓ, the change $c_i \to c_i + \delta$ would lead to the following change: $\boldsymbol{c}_B \to \boldsymbol{c}_B + \delta \boldsymbol{e}_\ell$.
  (1) *Condition for preserving optimality:* Since x_i is a basic variable, its reduced cost is always zero by [2.1.5]. Next, for every $j \neq i$, we can write

$$\begin{aligned} \bar{c}'_j &= \underbrace{c'_j}_{c_j} - \boldsymbol{A}_j^T (B^{-1})^T \underbrace{\boldsymbol{c}'_B}_{\boldsymbol{c}_B + \delta \boldsymbol{e}_\ell} \\ &= c_j - \boldsymbol{A}_j^T (B^{-1})^T \boldsymbol{c}_B - \delta \boldsymbol{A}_j^T (B^{-1})^T \boldsymbol{e}_\ell \\ &= \bar{c}_j - \delta \boldsymbol{e}_\ell^T B^{-1} \boldsymbol{A}_j, \end{aligned}$$

  and hence $\bar{c}'_j \geq 0 \iff \bar{c}_j \geq \delta \boldsymbol{e}_\ell^T B^{-1} \boldsymbol{A}_j$. Thus, the condition for preserving optimality is $\boxed{\bar{c}_j \geq \delta \boldsymbol{e}_\ell^T B^{-1} \boldsymbol{A}_j \ \forall j \neq i}$ where $\boldsymbol{e}_\ell^T B^{-1} \boldsymbol{A}_j$ is the ℓth entry of $B^{-1} \boldsymbol{A}_j$, i.e., the intersection of the jth column and ℓth row in the tableau from the simplex method:

$-\boldsymbol{c}_B^T \boldsymbol{x}_B$	$\cdots$	$\bar{c}_j$	$\cdots$
$\vdots$	$\vert$		$\vert$
$x_{B(\ell)}$	$\cdots$	$\boldsymbol{e}_\ell^T B^{-1} \boldsymbol{A}_j$	$\cdots$
$\vdots$	$\vert$		$\vert$

  (2) *Change in optimal value:* Since we have the following change: $\boldsymbol{c} \to \boldsymbol{c} + \delta \boldsymbol{e}_i$, the change in optimal value is $\boxed{\delta x_i}$.

4.1.3 **Case Study 2: Addition of a new variable.** With a new variable x_{n+1} added, the standard form LP problem becomes

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} + c_{n+1} x_{n+1} \\ \text{s.t.} \quad & A\boldsymbol{x} + \boldsymbol{A}_{n+1} x_{n+1} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, x_{n+1} \geq 0 \end{aligned}$$

Note that $(\boldsymbol{x}, x_{n+1}) = (\boldsymbol{x}^*, 0)$ is a basic feasible solution to the new LP problem with basis matrix B, where x_{n+1} is a nonbasic variable. Hence, if the basis matrix B remains optimal here, then $(\boldsymbol{x}, x_{n+1}) = (\boldsymbol{x}^*, 0)$ is an optimal solution to this new LP problem.

Since there are no changes to $c_1, \dots, c_n$ and B, we must have $B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$ and $\bar{\boldsymbol{c}} = \boldsymbol{c} - A^T (B^{-1})^T \boldsymbol{c}_B \geq \boldsymbol{0}$. So we only need to ensure that $\boxed{\bar{c}_{n+1} = c_{n+1} - \boldsymbol{c}_B^T B^{-1} \boldsymbol{A}_{n+1} \geq 0}$. When this condition is met, $(\boldsymbol{x}, x_{n+1}) = \boxed{(\boldsymbol{x}^*, 0)}$ is optimal, and the change in optimal value is $\boxed{0}$ since we have $c_{n+1} x_{n+1} = 0$ here.

4.2 Global Dependence on the Vectors $\boldsymbol{b}$ and $\boldsymbol{c}$

4.2.1 After studying *local* sensitivity analysis in Section 4.1, we will analyze the dependence of the optimal value on the vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ from a *global perspective*. First we discuss the global dependence on the vector $\boldsymbol{b}$.

4.2.2 **Setup.** To emphasize the dependence of the LP problem on the vector $\boldsymbol{b}$, we use the notation $P(\boldsymbol{b}) := \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ to denote the feasible set of the LP problem given the vector $\boldsymbol{b}$. Now, let $S := \{\boldsymbol{b} : P(\boldsymbol{b}) \text{ is nonempty}\} = \{\boldsymbol{Ax} : \boldsymbol{x} \geq \boldsymbol{0}\}$.

Then, for every $\boldsymbol{b} \in S$, we define $F(\boldsymbol{b}) := \min_{\boldsymbol{x} \in P(\boldsymbol{b})} \boldsymbol{c}^T\boldsymbol{x}$, which gives the optimal value of the LP problem, as a function of $\boldsymbol{b}$. By construction of S, we know $F(\boldsymbol{b}) < \infty$ always. To study the dependence of the LP problem on the vector $\boldsymbol{b}$, this function F on S is our main focus.

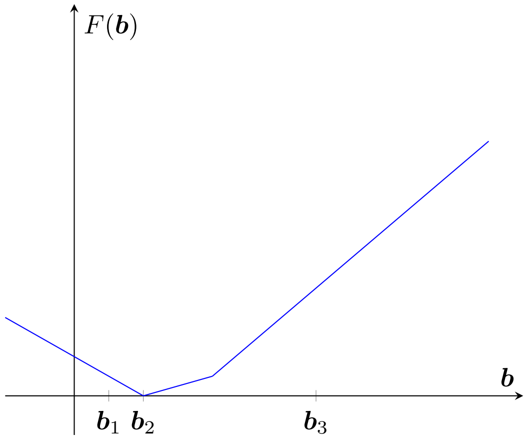
Throughout, we shall assume that the dual LP problem is feasible, i.e., $P^*(\boldsymbol{c}) := \{\boldsymbol{y} \in \mathbb{R}^m : A^T\boldsymbol{y} \leq \boldsymbol{c}\} \neq \varnothing$. In this case, by the weak duality we know that the primal optimal value must be greater than $-\infty$, meaning that $F(\boldsymbol{b})$ has to be finite for every $\boldsymbol{b} \in S$.

4.2.3 **Local analysis on the function F.** Before studying the function F globally, we first use the technique of local sensitivity analysis from Section 4.1 to understand better the properties of F.

First fix any $\boldsymbol{b}^* \in S$. With the vector $\boldsymbol{b} = \boldsymbol{b}^*$, suppose there is a nondegenerate primal optimal basic feasible solution, having the optimal basis matrix B. Then, the vector of basic variables is given by $\boldsymbol{x}_B = B^{-1}\boldsymbol{b}^* > \boldsymbol{0}$, where the positivity follows from the degeneracy. By Proposition 2.1.c, the reduced costs satisfy $\bar{\boldsymbol{c}} \geq \boldsymbol{0}$.

Now, suppose that we change $\boldsymbol{b}^* \to \boldsymbol{b}$. As long as the difference $\boldsymbol{b} - \boldsymbol{b}^*$ is sufficiently small, $B^{-1}\boldsymbol{b}$ would remain componentwise positive, establishing the *feasibility*. Also, this change would not affect $\bar{\boldsymbol{c}}$, so it remains nonnegative componentwise, establishing the *optimality*. This means that, provided that the change is sufficiently small, B would remain as an optimal basis. Therefore, we have $F(\boldsymbol{b}) = \boldsymbol{c}_B^T\boldsymbol{x}_B = \boldsymbol{c}_B^T B^{-1}\boldsymbol{b}$, for all $\boldsymbol{b}$ "near" $\boldsymbol{b}^*$.

Geometrically, this suggests that F is indeed a piecewise linear function of $\boldsymbol{b}$. To see this, note that $\frac{\partial F}{\partial \boldsymbol{b}} = (B^{-1})^T\boldsymbol{c}_B$ (by matrix calculus). So for all $\boldsymbol{b}$ "around" $\boldsymbol{b}^*$, F takes a constant "slope" as a function of $\boldsymbol{b}$. Of course, if $\boldsymbol{b}$ is sufficiently far from $\boldsymbol{b}^*$, the optimal basis matrix may change and so does $\frac{\partial F}{\partial \boldsymbol{b}}$, leading to a different "constant slope". This shows the piecewise linearity of F (under the assumption of the existence of such nondegenerate primal optimal basic feasible solution always).



In this picture, $\boldsymbol{b}_1$ and $\boldsymbol{b}_3$ yield nondegenerate primal optimal basic feasible solutions (they are candidates for "$\boldsymbol{b}^*$" here), while $\boldsymbol{b}_2$ cannot serve as a candidate for "$\boldsymbol{b}^*$" here, since F is not linear "around" $\boldsymbol{b}_2$; this suggests that $\boldsymbol{b}_2$ indeed yields a *degenerate* primal optimal basic feasible solution (knowing that every vector $\boldsymbol{b}$ must yield a primal optimal basic feasible solution, under the dual feasibility).

4.2.4 **Convexity of F.** Now we start studying F globally. The main property of F to be discussed here is its *convexity* as a function of $\boldsymbol{b}$:

Theorem 4.2.a. The function F is a convex function of $\boldsymbol{b}$ on the set S.

We will offer two proofs here: One is based on the definition; another is based on the duality theory.

Proof. (Definition) Fix any $\boldsymbol{b}_1, \boldsymbol{b}_2 \in S$ and any $\lambda \in [0,1]$. Since the optimal value of the standard form LP problem here is finite, it is always achieved at some point in the feasible region. Hence, we have $F(\boldsymbol{b}_1) = \boldsymbol{c}^T\boldsymbol{x}_1$ and $F(\boldsymbol{b}_2) = \boldsymbol{c}^T\boldsymbol{x}_2$ for some $\boldsymbol{x}_1 \in P(\boldsymbol{b}_1)$ and $\boldsymbol{x}_2 \in P(\boldsymbol{b}_2)$.

We claim that $\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in P(\lambda\boldsymbol{b}_1 + (1-\lambda)\boldsymbol{b}_2)$. To see this, consider:

- $A(\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) = \lambda A\boldsymbol{x}_1 + (1-\lambda)A\boldsymbol{x}_2 \overset{(\boldsymbol{x}_1\in P(\boldsymbol{b}_1),\boldsymbol{x}_2\in P(\boldsymbol{b}_2))}{=} \lambda\boldsymbol{b}_1 + (1-\lambda)\boldsymbol{b}_2$.
- From $\boldsymbol{x}_1, \boldsymbol{x}_2 \geq \boldsymbol{0}$, we have $\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \geq \boldsymbol{0}$.

Therefore, we have

$$\begin{aligned} F(\lambda\boldsymbol{b}_1 + (1-\lambda)\boldsymbol{b}_2) &= \min_{\boldsymbol{x}\in P(\lambda\boldsymbol{b}_1+(1-\lambda)\boldsymbol{b}_2)} \boldsymbol{c}^T\boldsymbol{x} \\ &\leq \boldsymbol{c}^T(\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) \\ &= \lambda\boldsymbol{c}^T\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{c}^T\boldsymbol{x}_2 \\ &= \lambda F(\boldsymbol{b}_1) + (1-\lambda)F(\boldsymbol{b}_2), \end{aligned}$$

establishing the convexity. □

Proof. (Duality theory) By strong duality, we know $F(\boldsymbol{b}) = \min_{\boldsymbol{x}\in P(\boldsymbol{b})} \boldsymbol{c}^T\boldsymbol{x} = \max_{\boldsymbol{y}\in P^*(\boldsymbol{c})} \boldsymbol{b}^T\boldsymbol{y}$, which is finite.

Claim: The dual feasible region $P^*(\boldsymbol{c}) = \{\boldsymbol{y} \in \mathbb{R}^m : A^T\boldsymbol{y} \leq \boldsymbol{c}\}$ has a vertex.

Proof. Since $P(\boldsymbol{b})$ is a nonempty standard form polyhedron and the optimal value is finite, applying the simplex method on the primal problem would yield an optimal basic feasible solution $\boldsymbol{x}$ with an optimal basis matrix B at the termination. Then, as in Lemma 3.3.a we can construct a basic solution $\boldsymbol{y} = (B^{-1})^T\boldsymbol{c}_B$ to the dual problem. Since the reduced costs must be all nonnegative at the termination of the simplex method, such basic solution $\boldsymbol{y}$ is also feasible for the dual problem. Hence it is a basic feasible solution, or vertex, of $P^*(\boldsymbol{c})$. □

Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ be the vertices of $P^*(\boldsymbol{c})$. By Theorem 1.2.f, the finite optimal value must be achieved at a vertex $\boldsymbol{y}_i$. So we can write $F(\boldsymbol{b}) = \max_{i=1,\ldots,N} \boldsymbol{b}^T\boldsymbol{y}_i$. As $\boldsymbol{b}^T\boldsymbol{y}_i$ is linear and hence convex in $\boldsymbol{b}$ for every $i = 1, \ldots, N$, we conclude by Proposition 1.1.c that F is convex in $\boldsymbol{b}$. □

4.2.5 **Global dependence on the vector $\boldsymbol{c}$.** So far we have been analyzing the global dependence of the vector $\boldsymbol{b}$. Next, we will consider the global dependence on the vector $\boldsymbol{c}$ instead. Likewise, we let $T := \{\boldsymbol{c} : P^*(\boldsymbol{c}) \text{ is nonempty}\}$ and define a function G on T by $G(\boldsymbol{c}) := \min_{\boldsymbol{x}\in P(\boldsymbol{b})} \boldsymbol{c}^T\boldsymbol{x} = \max_{\boldsymbol{y}\in P^*(\boldsymbol{c})} \boldsymbol{b}^T\boldsymbol{y}$ for every $\boldsymbol{c} \in T$, which always takes finite values.

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ be the vertices of the primal feasible region $P(\boldsymbol{b}) = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$. Then, using similar argument as before, we can write $G(\boldsymbol{c}) = \min_{i=1,\ldots,N} \boldsymbol{c}^T\boldsymbol{x}_i$, which is piecewise linear concave. (A function $f : \mathbb{R}^n \to \mathbb{R}$ is **concave** if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and all $\lambda \in [0,1]$, we have $f(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \geq \lambda f(\boldsymbol{x}) + (1-\lambda)f(\boldsymbol{y})$. Note that the negative of a concave function is convex.)

So, in some sense, the dependency of the optimal value on $\boldsymbol{c}$ (described by the function G) is somewhat "opposite" to that on $\boldsymbol{b}$ (described by the function F).

4.3 Parametric Programming

4.3.1 For the functions $F(\boldsymbol{b})$ and $G(\boldsymbol{c})$ introduced in Section 4.2, generally we can only have some *bounds* on their values:

- *(Lower bound)* $F(\boldsymbol{b}) \geq \boldsymbol{b}^T\boldsymbol{y}_i$ where $\boldsymbol{y}_i$ is any vertex of the dual feasible region $P^*(\boldsymbol{c})$.
- *(Upper bound)* $G(\boldsymbol{c}) \leq \boldsymbol{c}^T\boldsymbol{x}_i$ where $\boldsymbol{x}_i$ is any vertex of the primal feasible region $P(\boldsymbol{b})$.

In Section 4.3, we will study a special case where the *exact* optimal value can be efficiently determined as a function of a parameter θ that drives the changes on the vector $\boldsymbol{c}$ (hence the name parametric programming).

4.3.2 **Setup.** Again, we start by defining a function g by $g(\theta) := \min_{\boldsymbol{x}\in P(\boldsymbol{b})}(\boldsymbol{c}+\theta\boldsymbol{d})^T\boldsymbol{x}$ for all $\theta \in \mathbb{R}$. The function g represents the optimal value when the vector $\boldsymbol{c}$ is changed in a specific way, namely along a fixed direction $\boldsymbol{d}$. Here, we shall assume that the primal feasible set $P(\boldsymbol{b})$ is nonempty. Then, we know $g < \infty$, but it is still generally possible that $g(\theta) = -\infty$ for some $\theta \in \mathbb{R}$.

4.3.3 **Piecewise affinity and concavity of function g.** For every $\theta \in \mathbb{R}$ where $g(\theta)$ is finite (not $-\infty$), like before we can write $g(\theta) = \min_{i=1,\dots,N}(\boldsymbol{c}+\theta\boldsymbol{d})^T\boldsymbol{x}_i$ where $\boldsymbol{x}_1,\dots,\boldsymbol{x}_N$ are the vertices of the primal feasible region $P(\boldsymbol{b})$. From this, we can see that g is piecewise affine and concave.

4.3.4 **Computing $g(\theta)$ by the simplex method.** As mentioned earlier, the primary purpose of the function g is to allow us to efficiently compute the optimal value in respond to the change on the vector $\boldsymbol{c}$ along the direction $\boldsymbol{d}$. One way for performing such computation is through the (primal) simplex method, whose procedure is described as follows:

(1) Start with the tableau associated with a basic feasible solution $\boldsymbol{x}$ and the corresponding basis matrix B; here due to the presence of "θ" in the coefficients for the objective function, the *reduced cost* row of the tableau (zeroth row) would contain some θ's that are yet to be determined.

Example:

		x_1	x_2	x_3	x_4	x_5
	0	$-3+2\theta$	$3-\theta$	1	0	0
$x_4 =$	5	1	2	-3	1	0
$x_5 =$	7	2	1	-4	0	1

(2) Determine the bounds on θ such that all the reduced costs are nonnegative (so the current basic feasible solution is optimal), say $\ell \le \theta \le u$. Then, for every $\theta \in [\ell, u]$, the value of $g(\theta)$ is given by the negative of the top-left corner in the tableau.

Example: In the tableau above, all reduced costs are nonnegative iff $-3+2\theta \ge 0$ and $3-\theta \ge 0$, i.e., $3/2 \le \theta \le 3$. Hence, $g(\theta) = 0$ for every $\theta \in [3/2, 3]$.

(3) Consider the case where θ falls outside the bounds (slightly):

- *Case 1: $\theta > u$ (slightly).* Choose a column with negative reduced cost to be the pivot column (e.g., by Bland's pivoting rule). If no entry of $\boldsymbol{u}$ (the pivot column except the reduced cost entry) is positive, then we have $g(\theta) = -\infty$ for every $\theta > u$.

  If some entry of $\boldsymbol{u}$ is positive, choose the pivot row as usual, i.e., to be the one with the smallest ratio $x_{B(i)}/u_i$ over all i with $u_i > 0$. Then, perform the usual EROs on the tableau to change the basis.

  Example: In the case here, suppose $\theta > 3$ slightly. Then we would choose the second column to be the pivot column and the first row to be the pivot row:

		x_1	x_2	x_3	x_4	x_5
		$-3+2\theta$	$3-\theta$	1	0	0
$x_4 =$	5	1	$\boxed{2}$	-3	1	0
$x_5 =$	7	2	1	-4	0	1

  After performing the usual EROs (it is a bit tricker here due to the presence of θ), we get

		x_1	x_2	x_3	x_4	x_5
	$-7.5+2.5\theta$	$-4.5+2.5\theta$	0	$5.5-1.5\theta$	$-1.5+0.5\theta$	0
$x_4 =$	2.5	0.5	1	-1.5	0.5	0
$x_5 =$	4.5	1.5	0	-2.5	-0.5	1

- *Case 2: $\theta < \ell$ (slightly).* Choose a column with negative reduced cost to be the pivot column (e.g., by Bland's pivoting rule). If no entry of $\boldsymbol{u}$ is positive, then we have $g(\theta) = -\infty$ for every $\theta < \ell$.

  If some entry of $\boldsymbol{u}$ is positive, choose the pivot row as usual, and then perform the usual EROs on the tableau to change the basis.

(4) For each resulting tableau, go back to step 2 to determine the bounds on θ and the corresponding value of $g(\theta)$, and then proceed to step 3 again ↻...

Example: For the tableau obtained from the case with $\theta > 3$ slightly, the reduced costs are all nonnegative iff $3 \leq \theta \leq 5.5/1.5$. So we have $g(\theta) = -(-7.5 + 2.5\theta) = 7.5 - 2.5\theta$ for every $\theta \in [3, 5.5/1.5]$.

5 The Ellipsoid Method

5.0.1 Apart from the (dual) simplex method studied previously, there are also other algorithms for solving LP problems. A major motivation for developing these alternative algorithms is that while the simplex method is "often" efficient (and hence is indeed widely used for solving LP problems), it could take "exponential time" for some exceptional LP problems. On the other hand, the *ellipsoid method* (to be discussed in Section 5) provides a theoretical guarantee that LP problems can be solved in "polynomial time" (which is "faster" than "exponential time"). While the ellipsoid method is *not* an algorithm for solving LP problems by itself, such theoretical guarantee does lead to a new class of algorithms, known as *interior point methods*, which can solve LP problems efficiently; see Section 6.

5.0.2 The phrases "exponential time" and "polynomial time" mentioned above are used for describing the efficiency of algorithm. To understand them better, we will briefly study some related concepts in Section 5.1; you may consider taking a computer science course about algorithms (e.g., COMP2119) for a more detailed discussion on them.

5.1 Computational Complexity

5.1.1 **Basic terminologies.** Here, we shall focus on analyzing the computational complexity of algorithms for solving optimization problems, which describes how "fast" such algorithms are. We start by defining some related terms:

- An **instance** is a pair (S, f) where $S \subseteq \mathbb{R}^n$ is the feasible region and $f : S \to \mathbb{R}$ is the objective function.
- A **problem** is a collection of instances.
- An **algorithm** conducts a finite sequence of operations until a stopping criterion ⏸ is reached.
- The **size** of an instance is the number of parameters used to describe the instance, according to a prespecified format.
- The **running time** of an algorithm is the number of arithmetic operations needed until a stopping criterion ⏸ is reached, which generally depends on the instance.
- $T(n)$ denotes the worst-case (longest) running time of an algorithm over all instances of size n.
- An algorithm runs in **polynomial time** if there exists $k \in \mathbb{N}$ such that $T(n) = O(n^k)$, where we write $f(n) = O(g(n))$ if there exist $n_0 \in \mathbb{N}$ and $c > 0$ such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

  [Note: For our purpose here, we can treat an algorithm that takes exponential time as one with worst-case runtime growing exponentially in n.]

Since exponential growth is much quicker than polynomial growth, algorithms that run in polynomial time are regarded as "fast" (compared with algorithms that take exponential time at least).

Often, enumeration algorithms, i.e., algorithms that examine *every possible* candidate of optimal solutions, take exponential time. In the context of LP, an enumeration algorithm is to compute the objective function value for every vertex of the feasible region and then compare them. So, while such enumeration algorithms are quite easy to understand, they are rather inefficient most of the times.

5.1.2 **Iterative algorithms.** Many algorithms are *iterative*, whose operations are grouped into iterations. For example, the simplex method is an iterative algorithm. For this kind of algorithms, we have the following criterion for running in polynomial time.

Suppose that each iteration of an algorithm can be completed with *polynomially many operations* (with respect to the size of instance). Then, the algorithm runs in polynomial time iff the *number of iterations* (not operations) needed to solve an arbitrary instance of size n is $O(n^k)$.

5.1.3 **Simplex method takes exponential time.** As mentioned earlier, the simplex method is an iterative algorithm that takes exponential time. This fact is established by the following result, which gives us an example where the simplex method would take exponential time.

Proposition 5.1.a. Consider the LP problem

$$\begin{aligned} \min \quad & -x_n \\ \text{s.t.} \quad & \varepsilon \le x_1 \le 1-\varepsilon \\ & \varepsilon x_{i-1} \le x_i \le 1 - \varepsilon x_{i-1} \text{ for all } i = 2, \dots, n. \end{aligned}$$

(a) The feasible region has 2^n vertices.

(b) The vertices can be ordered such that each one is adjacent to and has lower objective function value than the previous one.

(c) There exists a pivoting rule where the simplex method requires 2^n iterations before it terminates.

[Intuition 💡: The feasible region is obtained by having *perturbations* on the unit cube which is defined by the constraints $0 \le x_1, \dots, x_n \le 1$. Here, we want to construct an example where the objective function value decreases *strictly* when moving a vertex to another (so that it would not be possible to "skip" many vertices and hence "quite many" iterations would be needed), and such perturbations can allow that.]

Proposition 5.1.a indicates that the simplex method would take exponential time, since for *some* pivoting rule, 2^n iterations are needed, which grow exponentially in n. However, it still leaves the possibility for the existence of some pivoting rule under which the simplex method can run in polynomial time; whether such pivoting rule exists is still an (important!) open question.

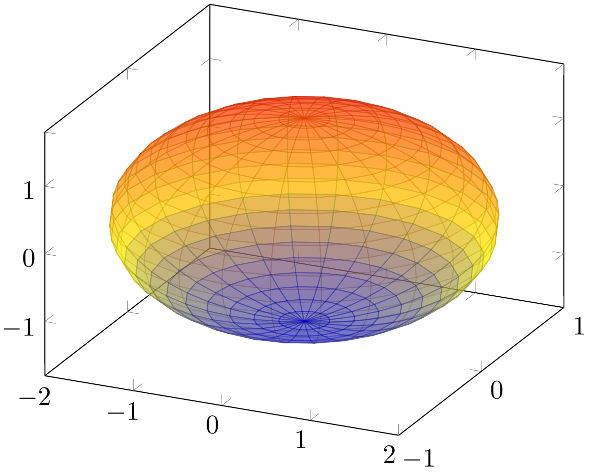
[Note: While the *worst-case* running time of the simplex method would be exponential time, in practice the simplex method is still quite efficient usually, and is a handy tool for solving LP problems.]

5.2 Development of The Ellipsoid Method

5.2.1 In view of the theoretical inefficiency of the simplex method (possibility of taking exponential time), next we will study the *ellipsoid method*, which can solve linear *feasibility* problems (i.e., deciding whether LP problems are feasible) with integer data in polynomial time. It serves as the building block for algorithms that *solve LP problems* in polynomial time (see Section 6).

5.2.2 **Geometry of the ellipsoid method.** We start by introducing geometric concepts related to the ellipsoid method:

- $E \subseteq \mathbb{R}^n$ is an **ellipsoid** (centered at $\boldsymbol{z}$) if it can be expressed as $E = E(\boldsymbol{z}, D) := \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x}-\boldsymbol{z})^T D^{-1}(\boldsymbol{x}-\boldsymbol{z}) \le 1\}$ where $\boldsymbol{z} \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n\times n}$ is positive definite.



Remarks:

– *(Positive definiteness)* A matrix $D \in \mathbb{R}^{n\times n}$ is **positive definite** if it is symmetric and we have $\boldsymbol{x}^T D\boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq \boldsymbol{0}$.

  - *(Special case — ball)* By taking $D = r^2 I$ with $r > 0$ (check that it is indeed positive definite!), the ellipsoid $E(\boldsymbol{z}, r^2 I)$ would become a *ball* centered at $\boldsymbol{z}$ with radius r. From this, we can see that the notion of *ellipsoid* is a generalization to *ball*. Indeed, ellipsoid can also be seen as a high-dimensional generalization to *ellipse* in $\mathbb{R}^2$.

- Let $D \in \mathbb{R}^{n \times n}$ be an invertible matrix and $\boldsymbol{b} \in \mathbb{R}^n$. Then the function $S : \mathbb{R}^n \to \mathbb{R}^n$ defined by $S(\boldsymbol{x}) := D\boldsymbol{x} + \boldsymbol{b}$ for all $\boldsymbol{x} \in \mathbb{R}^n$ is called an **affine transformation**. [Note: If $\boldsymbol{b} = \boldsymbol{0}$, then it becomes a *linear transformation* (the one from MATH2101). Here, the role of $\boldsymbol{b}$ is *translation* while the role of D is *scaling and rotation*.]
- Let $L \subseteq \mathbb{R}^n$. Then the **volume** of L is the multiple integral $\mathrm{Vol}(L) := \int_{\boldsymbol{x} \in L} \mathrm{d}\boldsymbol{x}$ (this is the same as the one from MATH2211).

5.2.3 **Properties about the geometrical concepts.** After learning the definitions of the geometrical concepts in [5.2.2], we will study some of their properties here.

(a) *(Characterization of positive definite matrix)* A matrix $D \in \mathbb{R}^{n \times n}$ is positive definite iff all of its eigenvalues are real and positive.

(b) *(Properties of positive definite matrices)* If $D \in \mathbb{R}^{n \times n}$ is positive definite, then:

i. *(Spectral theorem)* It has a diagonalization $D = P^{-1} \Lambda P$ where P is invertible and Λ is a diagonal matrix with positive diagonal entries.

ii. $\det D = \det \Lambda > 0$.

iii. D is invertible, and D^{-1} is positive definite.

iv. $D^{1/2} := P^{-1}\sqrt{\Lambda} P$ satisfies $D^{1/2} D^{1/2} = D$, and is invertible, where $\sqrt{\Lambda}$ is the diagonal matrix whose each diagonal entry is the square root of the corresponding one from Λ.

(c) Let $B := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^T \boldsymbol{x} \le 1\}$ be the unit ball centered at $\boldsymbol{0}$. Then, the ellipsoid $E(\boldsymbol{z}, D)$ can be expressed as $S(B)$ with the affine transformation $S(\boldsymbol{x}) = D^{1/2}\boldsymbol{x} + \boldsymbol{z}$.

(d) *(Volume formula after affine transformation)* Let $L \subseteq \mathbb{R}^n$ and $S(\boldsymbol{x}) = D\boldsymbol{x} + \boldsymbol{b}$ be an affine transformation. Then $\mathrm{Vol}(S(L)) = |\det D| \,\mathrm{Vol}(L)$.

[Note: In particular, this suggests that the volume of the ellipsoid $E(\boldsymbol{z}, D)$ is given by $\mathrm{Vol}(E(\boldsymbol{z}, D)) = \det D^{1/2}\,\mathrm{Vol}(B) = \sqrt{\det D}\,\mathrm{Vol}(B)$, where $\mathrm{Vol}(B)$ is the volume of the unit ball.]

Proof.

(a) Omitted.

(b) Omitted.

(c) Note that

$$\begin{aligned} S(B) &= \{\boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{y} = D^{1/2}\boldsymbol{x} + \boldsymbol{z}, \boldsymbol{x} \in B\} = \{\boldsymbol{y} \in \mathbb{R}^n : (D^{1/2})^{-1}(\boldsymbol{y} - \boldsymbol{z}) \in B\} \\ &= \{\boldsymbol{y} \in \mathbb{R}^n : (\boldsymbol{y} - \boldsymbol{z})^T (D^{1/2})^{-1} (D^{1/2})^{-1} (\boldsymbol{y} - \boldsymbol{z}) \le 1\} \\ &= \{\boldsymbol{y} \in \mathbb{R}^n : (\boldsymbol{y} - \boldsymbol{z})^T D^{-1} (\boldsymbol{y} - \boldsymbol{z}) \le 1\} = E(\boldsymbol{z}, D). \end{aligned}$$

(d) Using the change-of-variable formula for multiple integral, we have

$$\mathrm{Vol}(S(L)) = \int_{S(L)} \mathrm{d}\boldsymbol{y} = \int_L |\det J_S(\boldsymbol{x})| \,\mathrm{d}\boldsymbol{x} = \int_L |\det D| \,\mathrm{d}\boldsymbol{x} = |\det D| \,\mathrm{Vol}(L)$$

where $J_S(\boldsymbol{x})$ is the Jacobian matrix of S, which is given by $\begin{bmatrix} \frac{\partial S_1}{\partial x_1} & \cdots & \frac{\partial S_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial S_n}{\partial x_1} & \cdots & \frac{\partial S_n}{\partial x_n} \end{bmatrix} = D.$

□

5.2.4 **Solving linear feasibility problem by the ellipsoid method.** With the preparations above, we are now ready to study how to use the ellipsoid method for solving linear feasibility problem, i.e., given a polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$ with $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, determining whether P is empty. We start by introducing the intuitive idea of the ellipsoid method (which is perhaps more important than the actual implementation details!).

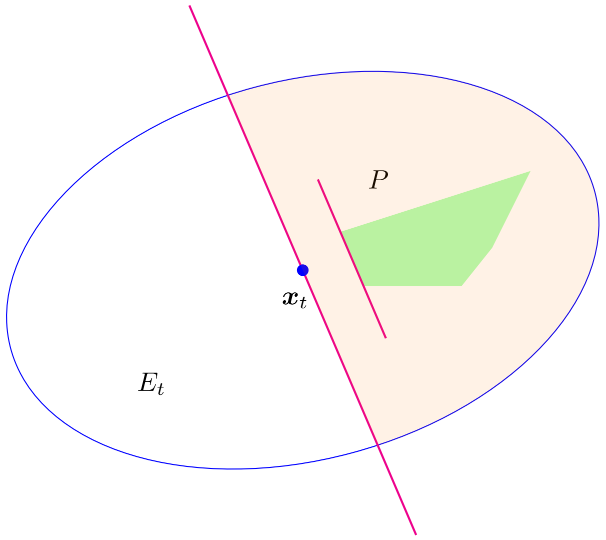
Intuition: The ellipsoid method generates a sequence of ellipsoids $\{E_t\}$ where the center of E_t is $\boldsymbol{x}_t$, such that $P \subseteq E_t$ for all t. Such sequence of ellipsoids is obtained in the following iterative process:

For each t:

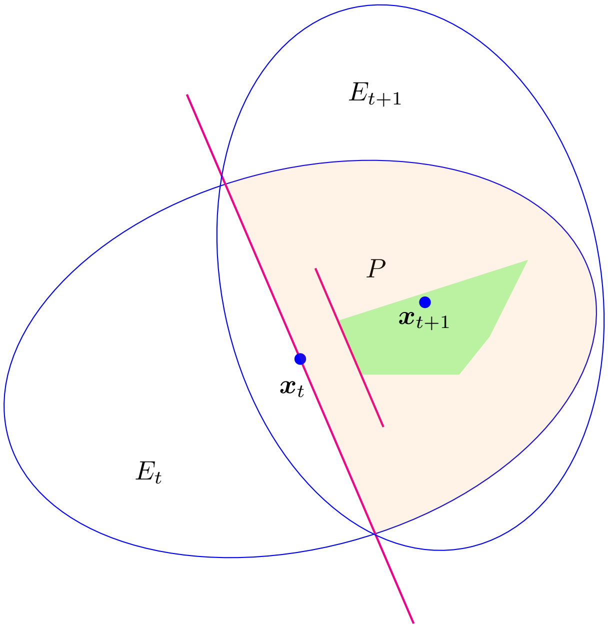
(a) If $\boldsymbol{x}_t \in P$, then we can immediately deduce that P is nonempty and we are done 👍.

(b) If $\boldsymbol{x}_t \notin P$, then there must exist a constraint that is violated, so we can write $\boldsymbol{a}^T\boldsymbol{x}_t < b$ where $\boldsymbol{a}^T$ is a row of A and b is the corresponding entry in $\boldsymbol{b}$. For every $\boldsymbol{x} \in P$, we then have $\boldsymbol{a}^T\boldsymbol{x} \geq b > \boldsymbol{a}^T\boldsymbol{x}_t$. Thus, P is a subset of the halfspace $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \geq \boldsymbol{a}^T\boldsymbol{x}_t\}$ which passes through $\boldsymbol{x}_t$ ("=" case), in addition to the ellipsoid E_t.

Geometrically, this means P is contained in the intersection of the ellipsoid E_t and a halfspace that passes through the center of E_t, which is said to be a **half-ellipsoid**.



Then, it turns out to be possible to find a new ellipsoid E_{t+1} that covers the half-ellipsoid, with volume being only a fraction of that of the previous ellipsoid E_t:



As the ellipsoids obtained in this way have shrinking volumes, eventually we would either find a point in P or find that the volume of P is so small that it must be empty (more details will be discussed in [5.2.6]). In either way, the linear feasibility problem is solved. This gives us an intuitive idea on how the ellipsoid method works.

Next, we will prove mathematical results that justify the ellipsoid method.

5.2.5 **Construction of those ellipsoids.** The first result to be established is an explicit way of constructing ellipsoids that satisfy the aforementioned properties. Particularly, we will provide a formula that computes "next ellipsoid" E_{t+1} given the ellipsoid E_k.

Theorem 5.2.a (Formula for "next ellipsoid")**.** *Let $n \geq 2$, $E = E(\boldsymbol{z}, D)$ be an ellipsoid in $\mathbb{R}^n$, and $\boldsymbol{a} \in \mathbb{R}^n$ be nonzero. Consider the halfspace $H := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \geq \boldsymbol{a}^T\boldsymbol{z}\}$. Let*

$$\bar{\boldsymbol{z}} = \boldsymbol{z} + \frac{1}{n+1} \cdot \frac{D\boldsymbol{a}}{\sqrt{\boldsymbol{a}^T D\boldsymbol{a}}} \quad \text{and} \quad \bar{D} = \frac{n^2}{n^2-1}\left(D - \frac{2}{n+1} \cdot \frac{D\boldsymbol{a}\boldsymbol{a}^T D}{\boldsymbol{a}^T D\boldsymbol{a}}\right).$$

Then $\bar{D}$ is positive definite and hence $E' := E(\bar{\boldsymbol{z}}, \bar{D})$ is an ellipsoid. Furthermore, we have $E \cap H \subseteq E'$ (containing the half-ellipsoid) and $\operatorname{Vol}(E') < e^{-\frac{1}{2(n+1)}} \operatorname{Vol}(E)$ (smaller volume).

[Note: In the special case where E is the unit ball (i.e., $\boldsymbol{z} = \boldsymbol{0}$ and $D = I$) with $\boldsymbol{a} = \boldsymbol{e}_1 = (1, 0, \ldots, 0)$, we would have $H = \{\boldsymbol{x} \in \mathbb{R}^n : x_1 \geq 0\}$,

$$\bar{\boldsymbol{z}} = \frac{1}{n+1} \cdot \boldsymbol{e}_1 \quad \text{and} \quad \bar{D} = \frac{n^2}{n^2-1}\left(I - \frac{2}{n+1} \cdot \boldsymbol{e}_1\boldsymbol{e}_1^T\right).$$

]

Proof. Here we only prove the special case where E is the unit ball (i.e., $\boldsymbol{z} = \boldsymbol{0}$ and $D = I$) with $\boldsymbol{a} = \boldsymbol{e}_1 = (1, 0, \ldots, 0)$. With the special case proved, the general case follows by performing suitable affine transformations; see Bertsimas and Tsitsiklis (1997, Theorem 8.1) for more details.

Showing the positive definiteness of $\bar{D}$. Note that

$$\bar{D} = \frac{n^2}{n^2-1}\left(\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} - \frac{2}{n+1}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}\right) = \frac{n^2}{n^2-1}\begin{bmatrix} \frac{n-1}{n+1} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Hence, for all $\boldsymbol{x} \neq \boldsymbol{0}$, we have

$$\boldsymbol{x}^T \bar{D} \boldsymbol{x} = \frac{n^2}{n-1}\left(\frac{n-1}{n+1}x_1^2 + \sum_{i=2}^{n} x_i^2\right) \overset{(x_i \neq 0 \text{ for some } i)}{>} 0.$$

Showing that $E \cap H \subseteq E'$**.** Since $\bar{D}$ is a diagonal matrix, we can compute its inverse conveniently:

$$\bar{D}^{-1} = \frac{n^2-1}{n^2}\begin{bmatrix} \frac{n+1}{n-1} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \frac{(n+1)^2}{n^2} & 0 & \cdots & 0 \\ 0 & \frac{n^2-1}{n^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{n^2-1}{n^2} \end{bmatrix}.$$

Hence, we have

$$\begin{aligned} E' &= \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x}-\boldsymbol{z})^T \bar{D}^{-1}(\boldsymbol{x}-\boldsymbol{z}) \leq 1\} \\ &= \left\{\boldsymbol{x} \in \mathbb{R}^n : \frac{(n+1)^2}{n^2}\left(x_1 - \frac{1}{n+1}\right)^2 + \frac{n^2-1}{n^2}\sum_{i=2}^{n} x_i^2 \leq 1\right\} \\ &= \left\{\boldsymbol{x} \in \mathbb{R}^n : \frac{(n+1)^2}{n^2}x_1^2 - \frac{2(n+1)}{n^2}x_1 + \frac{1}{n^2} + \frac{n^2-1}{n^2}\sum_{i=2}^{n} x_i^2 \leq 1\right\} \\ &= \left\{\boldsymbol{x} \in \mathbb{R}^n : \frac{n^2-1}{n^2}\sum_{i=1}^{n} x_i^2 + \frac{(n+1)^2-(n^2-1)}{n^2}x_1^2 - \frac{2(n+1)}{n^2}x_1 + \frac{1}{n^2} \leq 1\right\} \\ &= \left\{\boldsymbol{x} \in \mathbb{R}^n : \frac{n^2-1}{n^2}\sum_{i=1}^{n} x_i^2 + \frac{2(n+1)}{n^2}x_1(x_1-1) + \frac{1}{n^2} \leq 1\right\}. \end{aligned}$$

For all $\boldsymbol{x} \in E \cap H$, we have $\sum_{i=1}^{n} x_i^2 \leq 1$ and $x_1 \geq 0$, and thus

$$\frac{n^2-1}{n^2}\underbrace{\sum_{i=1}^{n} x_i^2}_{\leq 1} + \frac{2(n+1)}{n^2}x_1(x_1-1) + \frac{1}{n^2} \leq \frac{2(n+1)}{n^2}\underbrace{x_1}_{\geq 0}\underbrace{(x_1-1)}_{\leq 0} + 1 \leq 1,$$

meaning that $\boldsymbol{x} \in E'$.

Showing that $\operatorname{Vol}(E') < e^{-\frac{1}{2(n+1)}}\operatorname{Vol}(E)$**.** By [5.2.3], since E is the unit ball we know $\operatorname{Vol}(E') = \det \bar{D}^{1/2}\operatorname{Vol}(E) = \sqrt{\det \bar{D}}\operatorname{Vol}(E)$. Therefore, it suffices to show that $\sqrt{\det \bar{D}} < e^{-\frac{1}{2(n+1)}}$.

To show this, consider

$$\begin{aligned} \det \bar{D} &= \det\left(\frac{n^2}{n^2-1}\begin{bmatrix} \frac{n-1}{n+1} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}\right) = \left(\frac{n^2}{n^2-1}\right)^n \det\begin{bmatrix} \frac{n-1}{n+1} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \\ &= \left(\frac{n^2}{n^2-1}\right)^n \frac{n-1}{n+1} = \left(\frac{n^2}{n^2-1}\right)^{n-1}\frac{n^2}{n^2-1}\frac{n-1}{n+1} = \left(1 + \frac{1}{n^2-1}\right)^{n-1}\frac{n^2}{(n+1)^2} \\ &= \left(1 + \frac{1}{n^2-1}\right)^{n-1}\left(1 - \frac{1}{n+1}\right)^2 \overset{(1+x \leq e^x)}{\leq} \left(e^{\frac{1}{n^2-1}}\right)^{n-1}\left(e^{-\frac{1}{n+1}}\right)^2 = e^{-\frac{1}{n+1}}. \end{aligned}$$

Hence, we have $\sqrt{\det \bar{D}} \leq e^{-\frac{1}{2(n+1)}}$. □

5.2.6 **Implementation of the ellipsoid method.** Based on Theorem 5.2.a, we can implement the ellipsoid method for solving the linear feasibility problem. But here we need to impose some assumptions on the polyhedron P: (i) P is bounded, and (ii) P is either empty or **full-dimensional** (i.e., it has positive volume). [Note: It is possible to have a nonempty polyhedron with zero volume, e.g., a line in $\mathbb{R}^3$, but such polyhedron would have a lower dimension than the space. This explains the name "full-dimensional".]

With P being bounded, we know $P \subseteq E(\boldsymbol{x}_0, r^2 I)$ for some $\boldsymbol{x}_0 \in P$ and sufficiently large $r > 0$. Hence, we can set $V := \mathrm{Vol}\big(E(\boldsymbol{x}_0, r^2 I)\big)$ to serve as an initial upper bound on $\mathrm{Vol}(P)$. Also, in case P is full-dimensional, we would have $\mathrm{Vol}(P) > 0$, and thus $\mathrm{Vol}(P) \geq v$ for some $v > 0$. Altogether, we would have $v \leq \mathrm{Vol}(P) \leq V$.

Knowing $\boldsymbol{x}_0$, r, v, and V, the ellipsoid method can then be implemented as follows.

(1) *(Initialization)*

- $t^* \leftarrow \lceil 2(n+1)\ln(V/v) \rceil$.
- $E_0 \leftarrow E(\boldsymbol{x}_0, r^2 I)$.
- $D_0 \leftarrow r^2 I$.
- $t \leftarrow 0$.

(2) *(Main iteration)* Repeat the following steps until termination.

i. If $t = t^*$, conclude that P is empty and stop ⏸.

ii. If $\boldsymbol{x}_t \in P$, conclude that P is nonempty and stop ⏸.

iii. If $\boldsymbol{x}_t \notin P$, find a violated constraint with $\boldsymbol{a}_i^T \boldsymbol{x}_t < b_i$.

iv. Let $H_t := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_i^T \boldsymbol{x} \geq \boldsymbol{a}_i^T \boldsymbol{x}_t\}$. Construct the next ellipsoid E_{t+1} that contains $E_t \cap H_t$ and has a smaller volume than E_t as in Theorem 5.2.a:

$$\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + \frac{1}{n+1} \cdot \frac{D_t \boldsymbol{a}_i}{\sqrt{\boldsymbol{a}_i^T D_t \boldsymbol{a}_i}},$$

$$D_{t+1} \leftarrow \frac{n^2}{n^2-1}\left(D_t - \frac{2}{n+1} \cdot \frac{D_t \boldsymbol{a}_i \boldsymbol{a}_i^T D_t}{\boldsymbol{a}_i^T D_t \boldsymbol{a}_i}\right).$$

v. $t \leftarrow t+1$.

Remarks:

- The case where $t = t^*$ corresponds to the conclusion that "the volume of P is so small that it must be empty" mentioned in the intuitive idea. See the proof of Theorem 5.2.b for more details about this.
- By induction, we can show that $P \subseteq E_t$ for every t in the ellipsoid method.
  First, by construction we have $P \subseteq E_0$. Next, assume $P \subseteq E_k$ for some $k < t^*$. The existence of E_k implies that there is a violated constraint $\boldsymbol{a}_i^T \boldsymbol{x}_k < b_i$. Then, for all $\boldsymbol{x} \in P$, we have $\boldsymbol{a}_i^T \boldsymbol{x} \geq b_i > \boldsymbol{a}_i^T \boldsymbol{x}_k$, implying that $P \subseteq H_k$, and hence $P \subseteq E_k \cap H_k \overset{\text{(Theorem 5.2.a)}}{\subseteq} E_{k+1}$. This completes the proof by induction.

5.2.7 **Correctness of the ellipsoid method.** From the implementation of the ellipsoid method in [5.2.6], while we know that it must terminate with a conclusion that P is empty/nonempty by construction, we actually have not shown that such conclusion is *correct*. Particularly, it may be possible that when $t = t^*$, we conclude that P is empty but it is indeed not. [Note: On the other hand, it is not possible to wrongly conclude that P is nonempty, since $\boldsymbol{x}_t \in P$ does imply that P is nonempty.]

To fully justify the ellipsoid method, we need the following result that guarantees its correctness.

Theorem 5.2.b (The ellipsoid method is correct)**.** Let P be a bounded polyhedron that is either empty or full-dimensional. The ellipsoid method from [5.2.6] correctly determines whether P is empty, i.e., if $\boldsymbol{x}_t \notin P$ for all $t = 0, 1, \dots, t^*-1$ (so we would stop at the case where $t = t^*$), then P is empty.

Proof. Suppose $\boldsymbol{x}_t \notin P$ for all $t = 0, 1, \dots, t^*-1$. Then, by Theorem 5.2.a we have

$$\mathrm{Vol}(E_{t^*}) < e^{-\frac{t^*}{2(n+1)}} \mathrm{Vol}(E_0) \leq V e^{-\frac{2(n+1)\ln(V/v)}{2(n+1)}} = V \cdot \frac{v}{V} = v.$$

Since $P \subseteq E_{t^*}$, we have $\mathrm{Vol}(P) \leq \mathrm{Vol}(E_{t^*}) < v$, which implies that P is not full-dimensional, and hence empty. □

5.2.8 **Relaxing the assumptions on the boundedness and full-dimensionality.** In the ellipsoid method, we require the boundedness and full-dimensionality assumptions, but in practice the polyhedron P in consideration may violate one of these assumptions. In view of this, here we will discuss a method that allows such assumptions to be relaxed, but requires us to further assume that all the entries of A and $\boldsymbol{b}$ for the polyhedron P are integers *(integer data)*.

The basic idea of the method is as follows. Given any polyhedron P (possibly violating the assumptions on boundedness and full-dimensionality), we can construct another polyhedron P' such that (i) P is nonempty iff P' is nonempty (so we can check whether P' is empty rather than P), and (ii) if P' is nonempty, then $v \leq \mathrm{Vol}(P') \leq V$ for some $v, V > 0$; here, the upper and lower bounds on $\mathrm{Vol}(P')$ correspond to the boundedness and full-dimensionality assumptions respectively. Then, we can apply the ellipsoid method on P' to deduce whether P is empty.

5.2.9 **Relaxing the boundedness assumption.** To relax the boundedness assumption, we need to establish a result that bounds every vertex of the polyhedron P with integer data.

Proposition 5.2.c. Let $A \in \mathbb{Z}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, and $U \in \mathbb{N}$ be an upper bound on the absolute values of the entries in A and $\boldsymbol{b}$.[7] Suppose that the polyhedron P in consideration has a vertex.

(a) *(General form)* Every vertex $\boldsymbol{x} = (x_1, \dots, x_n)$ of the polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$ satisfies $|x_j| \leq (nU)^n$ for every $j = 1, \dots, n$.

(b) *(Standard form)* Every vertex $\boldsymbol{x} = (x_1, \dots, x_n)$ of the polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ satisfies $|x_j| \leq (mU)^m$ for every $j = 1, \dots, n$.

Proof.

(a) Fix any vertex $\boldsymbol{x}$ of P. Since $\boldsymbol{x}$ is also a basic feasible solution of P, we can choose n linearly independent active constraints at $\boldsymbol{x}$, and then write $\widehat{A}\boldsymbol{x} = \widehat{\boldsymbol{b}}$, where $\widehat{A}$ is an $n \times n$ invertible submatrix of A and $\widehat{\boldsymbol{b}}$ is the corresponding n-dimensional subvector of $\boldsymbol{b}$. By Cramer's rule, the solution $\boldsymbol{x}$ to $\widehat{A}\boldsymbol{x} = \widehat{\boldsymbol{b}}$ is given by $x_j = \det \widehat{A}^j / \det \widehat{A}$ for all $j = 1, \dots, n$, where $\widehat{A}^j$ denotes the matrix $\widehat{A}$ with its jth column replaced by $\widehat{\boldsymbol{b}}$.
Using the Leibniz formula for determinants, we have $\det \widehat{A}^j = \sum_\sigma (-1)^{|\sigma|} \prod_{i=1}^n \widetilde{a}_{i,\sigma(i)}$, where the summation is over all the $n!$ permutations $\sigma = (\sigma(1), \dots, \sigma(n))$ of $\{1, \dots, n\}$, with $\widetilde{a}_{ij} = \widehat{b}_j$ and $\widetilde{a}_{ik} = \widehat{a}_{ik}$ for all $k \neq j$; here $|\sigma|$ denotes the number of *inversions* of the permutation σ, i.e., number of times where $i < j$ while $\sigma(i) > \sigma(j)$.
Then, consider

$$|\det \widehat{A}^j| = \left|\sum_\sigma (-1)^{|\sigma|} \prod_{i=1}^n \widetilde{a}_{i,\sigma(i)}\right| \overset{\text{(triangle)}}{\leq} \sum_\sigma \prod_{i=1}^n |\widetilde{a}_{i,\sigma(i)}|$$
$$\overset{(|\widetilde{a}_{i,\sigma(i)}|\leq U)}{\leq} \sum_\sigma U^n = n!U^n \overset{(n!=n(n-1)\cdots 1\leq n(n)\cdots n=n^n)}{\leq} (nU)^n$$

for all $j = 1, \dots, n$. Furthermore, since $\det \widehat{A} \neq 0$ (as $\widehat{A}$ is invertible) and $\widehat{A}$ has integer entries, we must have $|\det \widehat{A}| \geq 1$. Hence, for all $j = 1, \dots, n$, we have $|x_j| = |\det \widehat{A}^j| / |\det \widehat{A}| \leq (nU)^n / 1 = (nU)^n$.

(b) Use the same argument as in (a) but replace the $n \times n$ matrix $\widehat{A}$ by the basis matrix B. In case A has linearly independent rows, the basis matrix would be of size $m \times m$. Otherwise, we can remove redundant rows of A and so the size of the basis matrix would be smaller than $m \times m$. Thus, in either case, repeating the argument in (a) would at least yield the bound $|x_j| \leq (mU)^m$ for every $j = 1, \dots, n$.

[7] Such U always exists as there are only finitely many entries here.

□

To use Proposition 5.2.c for relaxing the boundedness assumption, we can do the following. First we start with any polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$, where the rows of A span $\mathbb{R}^n$ (implying that $m \geq n$). Then, define

$$P_B := \{\boldsymbol{x} \in P : |x_j| \leq (nU)^n \ \forall j = 1, \dots, n\}.$$

Here, the polyhedron P_B satisfies the following properties:

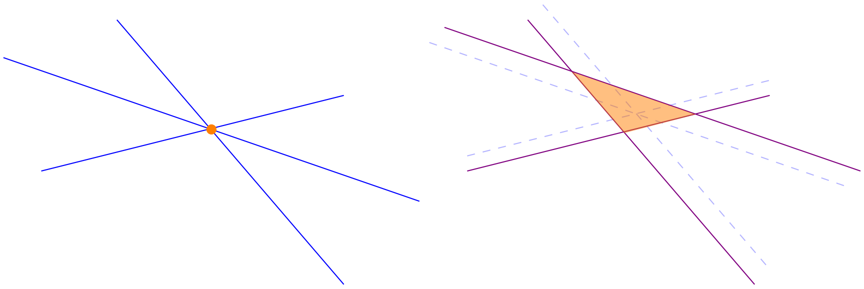
(a) P is nonempty iff P_B is nonempty.

[Note: This holds under the assumption that the rows of A span $\mathbb{R}^n$, which would mean that P is nonempty iff P has a vertex iff P_B is nonempty.]

(b) P_B is a bounded polyhedron contained in the ball $E_0 := E(\boldsymbol{0}, r^2 I) := E(\boldsymbol{0}, n(nU)^{2n} I)$.

[Note: The volume of E_0 is less than $(2r)^n = (2n(nU)^n)^n = (2n)^n (nU)^{n^2}$, so we can take $V = (2n)^n (nU)^{n^2}$ as the initial upper bound of $\mathrm{Vol}(P_B)$ in the ellipsoid method.]

5.2.10 **Relaxing the full-dimensionality assumption.** Next, we will discuss how the full-dimensionality assumption can be relaxed. Similar to [5.2.9], a result is needed for constructing a polyhedron that is full-dimensional from a polyhedron P. The key idea of this result is that a small perturbation of a nonempty polyhedron P (possibly not full-dimensional) can yield a full-dimensional polyhedron P_ε.



Proposition 5.2.d. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$ be a polyhedron with $A \in \mathbb{Z}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, and $U \in \mathbb{N}$ being an upper bound on the absolute values of the entries in A and $\boldsymbol{b}$. We let

$$\varepsilon = \frac{1}{2(n+1)}((n+1)U)^{-(n+1)} \quad \text{and} \quad P_\varepsilon = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b} - \varepsilon\boldsymbol{1}\}$$

where $\boldsymbol{1} = (1, \dots, 1) \in \mathbb{R}^m$.

(a) If P is empty, then P_ε is empty.

(b) If P is nonempty, then P_ε is full-dimensional.

Proof.

(a) Assume P is empty.

Setting up two pairs of primal and dual. Then consider the following infeasible (primal) LP problem:

$$\begin{aligned} \min \quad & \boldsymbol{0}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \end{aligned}$$

and its dual:

$$\begin{aligned} \max \quad & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} \quad & A^T\boldsymbol{y} = \boldsymbol{0} \\ & \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

Since the primal problem is infeasible, by [3.2.4] we know that either the dual problem is infeasible or its optimal value is ∞. Observing that $\boldsymbol{y} = \boldsymbol{0}$ is a feasible solution to the dual, we deduce that its optimal value must be ∞. So particularly, there exists $\boldsymbol{y} \in \mathbb{R}^m$ such that $A^T\boldsymbol{y} = \boldsymbol{0}$ and $\boldsymbol{b}^T\boldsymbol{y} = 1$.

Also, consider the "perturbed" primal:

$$\begin{aligned} \min \quad & \boldsymbol{0}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} - \varepsilon\boldsymbol{1}, \end{aligned}$$

and the "perturbed" dual:

$$\begin{aligned} \max \quad & (\boldsymbol{b} - \varepsilon\boldsymbol{1})^T\boldsymbol{y} \\ \text{s.t.} \quad & A^T\boldsymbol{y} = \boldsymbol{0} \\ & \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

Showing the existence of feasible solution to the "perturbed" dual with positive objective function value. Let $P_1 = \{\boldsymbol{y} \in \mathbb{R}^m : A^T\boldsymbol{y} = \boldsymbol{0}, \boldsymbol{b}^T\boldsymbol{y} = 1, \boldsymbol{y} \geq \boldsymbol{0}\}$, which is a nonempty standard form polyhedron. By Corollary 1.2.e, P_1 has a vertex and hence by Proposition 5.2.c, there exists a basic feasible solution $\widehat{\boldsymbol{y}}$ of P_1 such that $|\widehat{y}_j| \leq ((n+1)U)^{n+1}$ $\forall j = 1, \ldots, m$ (the "m" for that result is $n+1$ here). Since $\widehat{\boldsymbol{y}}$ is a basic feasible solution, at most $n+1$ of its components are nonzero, and thus $\boldsymbol{1}^T\widehat{\boldsymbol{y}} = \sum_{j=1}^m \widehat{y}_j \leq (n+1)((n+1)U)^{n+1}$. This implies that

$$(\boldsymbol{b} - \varepsilon\boldsymbol{1})^T\widehat{\boldsymbol{y}} = \boldsymbol{b}^T\widehat{\boldsymbol{y}} - \varepsilon\boldsymbol{1}^T\boldsymbol{y} \geq 1 - \frac{1}{2(n+1)}((n+1)U)^{-(n+1)}(n+1)((n+1)U)^{n+1} = 1 - \frac{1}{2} > 0,$$

and hence such $\widehat{\boldsymbol{y}}$ is a feasible solution to the "perturbed" dual with positive objective function value.

Completing the proof by showing that the optimal value of "perturbed" dual is ∞. Fix any $\lambda > 0$. Since $A^T(\lambda\widehat{\boldsymbol{y}}) = \boldsymbol{0}$ and $\lambda\widehat{\boldsymbol{y}} \geq \boldsymbol{0}$, $\lambda\widehat{\boldsymbol{y}}$ is a feasible solution to the "perturbed" dual. Now, note that the objective function at $\lambda\widehat{\boldsymbol{y}}$ is $(\boldsymbol{b} - \varepsilon\boldsymbol{1})^T(\lambda\widehat{\boldsymbol{y}}) = \lambda\underbrace{(\boldsymbol{b} - \varepsilon\boldsymbol{1})^T\widehat{\boldsymbol{y}}}_{>0}$, which can be arbitrarily large as λ is increased, so we conclude that the optimal value of "perturbed" dual is ∞. Hence, by [3.2.4], we know that the "perturbed" primal must be infeasible, implying that P_ε is empty.

(b) Assume P is nonempty and let $\boldsymbol{x} \in P$, which satisfies that $A\boldsymbol{x} \geq \boldsymbol{b}$. Let $C := \{\boldsymbol{z} \in \mathbb{R}^n : |z_j - x_j| \leq \varepsilon/(nU)\ \forall j = 1, \ldots, n\}$, which is a cube with positive volume.

Fix any $\boldsymbol{z} \in C$. Note that we have for all $i = 1, \ldots, m$,

$$\boldsymbol{a}_i^T\boldsymbol{z} = \boldsymbol{a}_i^T\boldsymbol{x} + \underbrace{\boldsymbol{a}_i^T(\boldsymbol{z} - \boldsymbol{x})}_{\sum_{j=1}^n a_{ij}(z_j - x_j)} \geq b_i - \sum_{j=1}^n |a_{ij}| \cdot |z_j - x_j| \geq b_i - \sum_{j=1}^n U \cdot \varepsilon/(nU) = b_i - \varepsilon,$$

which implies that $\boldsymbol{z} \in P_\varepsilon$. Hence, $C \subseteq P_\varepsilon$, and thus $\text{Vol}(P_\varepsilon) \geq \text{Vol}(C) > 0$, meaning that P_ε is full-dimensional.

□

From Proposition 5.2.d, we know that the perturbed polyhedron P_ε satisfies:

(a) P is nonempty iff P_ε is nonempty.

(b) P_ε is either empty or full-dimensional (as P is either empty or nonempty).

Therefore, it can be used to relax the full-dimensionality assumption. Furthermore, we can apply the method from [5.2.9] to obtain a bounded and full-dimensional polyhedron that is nonempty iff the original polyhedron is nonempty (more details in [5.2.11]). However, to perform the ellipsoid method on such polyhedron, we would need the input v, which is not given in Proposition 5.2.d. The following result addresses this need and provides a possible value of v.

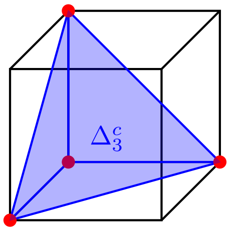
Proposition 5.2.e. Let $A \in \mathbb{Z}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, and $U \in \mathbb{N}$ be an upper bound on the absolute values of the entries in A and $\boldsymbol{b}$. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$ be a bounded and full-dimensional polyhedron. We have

$$\mathrm{Vol}(P) > n^{-n}(nU)^{-n^2(n+1)}$$

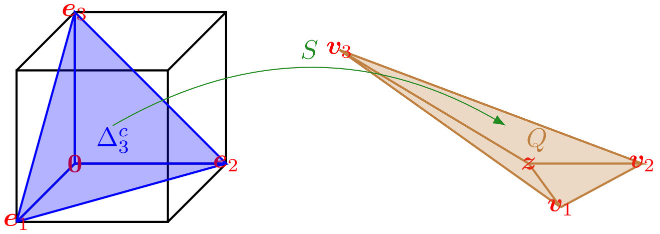
(hence one possible choice of v is $n^{-n}(nU)^{-n^2(n+1)}$).

Proof. Here, we will use the fact that a nonempty and bounded polyhedron is the *convex hull* of its vertices $\boldsymbol{v}^0, \dots, \boldsymbol{v}^N$, given by $\mathrm{conv}\big(\{\boldsymbol{v}^0, \dots, \boldsymbol{v}^N\}\big) := \{\sum_{i=0}^N \lambda_i \boldsymbol{v}^i : \sum_{i=0}^N \lambda_k = 1, \lambda_i \geq 0\ \forall i\}$.

With P being full-dimensional, it can be shown that there are $n+1$ vertices of P that are not on a common hyperplane, say $\boldsymbol{v}^0, \dots, \boldsymbol{v}^n$ without loss of generality. Now, we let $Q = \mathrm{conv}\big(\{\boldsymbol{v}^0, \dots, \boldsymbol{v}^n\}\big) \subseteq P$, and let $\Delta_n^c := \{\boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq 1, x_i \geq 0\ \forall i\}$ be a *corner of cube* for each $n \in \mathbb{N}$, whose volume can be shown to be $1/n!$.



Next, we will use the fact that Q can be obtained by applying an affine transformation S on Δ_n^c, with S characterized by the mappings for vertices: $S(\boldsymbol{e}_i) = \boldsymbol{v}^i$ for all $i = 0, \dots, n$, where $\boldsymbol{e}_0 := \boldsymbol{0}$.



By writing the affine transformation as $S(\boldsymbol{x}) = D\boldsymbol{x} + \boldsymbol{z}$, we can deduce the expressions of D and $\boldsymbol{z}$ as follows:

- $S(\boldsymbol{0}) = \boldsymbol{z} = \boldsymbol{v}^0$.
- For all $i = 1, \dots, n$, $S(\boldsymbol{e}_i) = D\boldsymbol{e}_i + \boldsymbol{z} = \boldsymbol{v}^i \implies D\boldsymbol{e}_i = \boldsymbol{v}^i - \boldsymbol{v}^0$.

Therefore, we have $\boldsymbol{z} = \boldsymbol{v}^0$ and $D = \begin{bmatrix}\boldsymbol{v}^1 - \boldsymbol{v}^0 & \cdots & \boldsymbol{v}^n - \boldsymbol{v}^0\end{bmatrix}$, which means that

$$Q = \begin{bmatrix}\boldsymbol{v}^1 - \boldsymbol{v}^0 & \cdots & \boldsymbol{v}^n - \boldsymbol{v}^0\end{bmatrix}\Delta_n^c + \boldsymbol{v}^0.$$

Now, consider

$$\det D = \begin{vmatrix}\boldsymbol{v}^1 - \boldsymbol{v}^0 & \cdots & \boldsymbol{v}^n - \boldsymbol{v}^0\end{vmatrix} \overset{\text{(cofactor expansion)}}{=} \begin{vmatrix}1 & 0 & \cdots & 0 \\ \boldsymbol{v}^0 & \boldsymbol{v}^1 - \boldsymbol{v}^0 & \cdots & \boldsymbol{v}^n - \boldsymbol{v}^0\end{vmatrix}$$

$$\overset{\text{(column operations)}}{=} \begin{vmatrix}1 & 1 & \cdots & 1 \\ \boldsymbol{v}^0 & \boldsymbol{v}^1 & \cdots & \boldsymbol{v}^n\end{vmatrix} = \begin{vmatrix}1 & 1 & \cdots & 1 \\ v_1^0 & v_1^1 & \cdots & v_1^n \\ \vdots & \vdots & \ddots & \vdots \\ v_n^0 & v_n^1 & \cdots & v_n^n\end{vmatrix} \underset{(\sigma\text{ is permutation of }\{0,\dots,n\})}{\overset{\text{(Leibniz)}}{=}} \sum_\sigma (-1)^{|\sigma|} 1 \cdot \prod_{i=1}^n v_i^{\sigma(i)}.$$

By Cramer's rule, we know for all $i, k = 1, \dots, n$, the value v_i^k is a quotient of determinants, which are integers due to the integer data assumption, with denominator being nonzero. Using a similar argument as in the proof of Proposition 5.2.c, we can show that the absolute values of determinants can all be bounded above by $(nU)^n$. Furthermore, since D is invertible and thus $\det D \neq 0$, the numerators must be all nonzero as well. Hence, we can write $v_i^k = p_i^k/q_i^k$, with $1 \leq |p_i^k|, |q_i^k| \leq (nU)^n$, for all $i, k = 1, \dots, n$.

Consequently, we have

$$\det D = \sum_{\sigma} (-1)^{|\sigma|} \prod_{i=1}^{n} v_i^{\sigma(i)} = \sum_{\sigma} (-1)^{|\sigma|} \frac{\prod_{i=1}^{n} p_i^{\sigma(i)}}{\prod_{i=1}^{n} q_i^{\sigma(i)}} \overset{\text{(summing fractions)}}{\underset{\text{(common denominator)}}{=}} \frac{a}{\prod_{i=1}^{n}\prod_{k=0}^{n} q_i^k}$$

where $a \in \mathbb{Z} \setminus \{0\}$. Thus,

$$|\det D| = \left| \frac{a}{\prod_{i=1}^{n}\prod_{k=0}^{n} q_i^k} \right| \geq \frac{1}{\prod_{i=1}^{n}\prod_{k=0}^{n} |q_i^k|} \geq \frac{1}{((nU)^n)^{n(n+1)}} = \frac{1}{(nU)^{n^2(n+1)}}.$$

Hence,

$$\mathrm{Vol}(Q) = |\det D| \, \mathrm{Vol}(\Delta_n^c) \geq \frac{1}{(nU)^{n^2(n+1)}} \cdot \frac{1}{n!} \geq n^{-n}(nU)^{-n^2(n+1)}.$$

□

5.2.11 **Implementing the ellipsoid method with polynomial runtime.** We now have enough ingredients to demonstrate that the ellipsoid method can run in polynomial time. It is stated more formally in the result below:

Theorem 5.2.f. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{b}\}$ be a polyhedron with $A \in \mathbb{Z}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, and $U \in \mathbb{N}$ being an upper bound on the absolute values of the entries in A and $\boldsymbol{b}$. The linear feasibility problem of determining whether P is empty can be solved in polynomial time.

Proof. We will solve this problem by the ellipsoid method. By [5.1.2] and the fact that each iteration in the ellipsoid method can be completed with polynomially many operations (it indeed takes some work to establish this), it suffices to show that its number of iterations needed would be $O(n^k)$ for some $k \in \mathbb{N}$.

From [5.2.6] we know that the number of iterations needed for the ellipsoid method is at most $t^* = \lceil 2(n+1)\ln(V/v) \rceil$. With $v = n^{-n}(nU)^{-n^2(n+1)}$ and $V = (2n)^n(nU)^{n^2}$ (these expressions come from [5.2.9] and [5.2.10]), we have

$$\begin{aligned} t^* &= \left\lceil 2(n+1)\ln \frac{(2n)^n(nU)^{n^2}}{n^{-n}(nU)^{-n^2(n+1)}} \right\rceil \leq 2(n+1)\ln\left((2n)^{2n}(nU)^{n^2(n+2)}\right) + 1 \\ &= 2(n+1)\cdot 2n\ln(2n) + n^2(n+2)\ln(nU) \overset{\text{(focus on "highest order" term)}}{=} O(n^4\ln(nU)), \end{aligned}$$

so the number of iterations needed would be $O(n^4 \ln(nU))$ for such case.

Given the polyhedron P, according to [5.2.9] we can form a bounded polyhedron $P_B = \{\boldsymbol{x} \in P : |x_j| \leq (nU)^n \ \forall j = 1, \dots, n\} = \{\boldsymbol{x} \in \mathbb{R}^n : A^*\boldsymbol{x} \geq \boldsymbol{b}^*\}$. For the polyhedron P_B, the entries in the vector $\boldsymbol{b}^*$ would include the integers $(nU)^n$ and $-(nU)^n$, and so we would need to change the upper bound in absolute value on the integer entries from U to $(nU)^n$.

Next, according to [5.2.10], we can perturb P_B as in Proposition 5.2.d to form a new polyhedron $P_{B,\varepsilon} = \{\boldsymbol{x} \in \mathbb{R}^n : A^*\boldsymbol{x} \geq \boldsymbol{b}^* - \varepsilon\mathbf{1}\} = \{\boldsymbol{x} \in \mathbb{R}^n : (1/\varepsilon)A^*\boldsymbol{x} \geq (1/\varepsilon)\boldsymbol{b}^* - \mathbf{1}\}$ with $\varepsilon = \frac{1}{2(n+1)}((n+1)(nU)^n)^{-(n+1)}$. Note that $1/\varepsilon = 2(n+1)((n+1)(nU)^n)^{n+1}$ is an integer, and for the polyhedron $P_{B,\varepsilon}$, we would need to change the upper bound in absolute value on the integer entries from $(nU)^n$ to $(1/\varepsilon)(nU)^n + 1$, which can be further bounded as follows:

$$(1/\varepsilon)(nU)^n + 1 = 2(n+1)((n+1)(nU)^n)^{n+1}(nU)^n + 1 \leq 2(n+1)^{3n+6}U^{n(n+2)},$$

so we may update the upper bound to $2(n+1)^{3n+6}U^{n(n+2)}$ instead for the polyhedron $P_{B,\varepsilon}$.

Then, we can apply the ellipsoid method on $P_{B,\varepsilon}$ with v and V being the ones from above, but with $U \to 2(n+1)^{3n+6}U^{n(n+2)}$, which yields

$$\begin{aligned} t^* &= O(n^4 \ln(n \cdot 2(n+1)^{3n+6}U^{n(n+2)})) = O(n^4(\ln n + (3n+6)\ln(2n+2) + n(n+2)\ln U)) \\ &\overset{\text{(drop lower order terms)}}{=} O(n^4(3n\ln n + n^2 \ln U)) \overset{\text{(consider definition)}}{=} O(n^4(3n^2 \ln nU + n^2 \ln nU)) \\ &\overset{\text{(consider definition)}}{=} O(n^4 n^2 \ln(nU)) = O(n^6 \ln(nU)). \end{aligned}$$

So, the number of iterations needed in general would be $O(n^6 \ln nU)$. Furthermore, since $\ln nU \leq nU$, we can indeed write it as $O(n^6(nU)) = O(n^7)$, as desired. □

5.2.12 **Solving LPs with integer data by the ellipsoid method in polynomial time.** Previously, we have only applied the ellipsoid method for solving linear *feasibility* problems, and it has been shown to be a method for doing that in polynomial time with integer data. It turns out that the ellipsoid method can again be utilized for solving actual LP problems (again with integer data) in polynomial time. The basic idea is as follows.

Consider a standard form primal LP

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

and its dual

$$\begin{aligned} \max \quad & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} \quad & A^T\boldsymbol{y} = \boldsymbol{c}, \end{aligned}$$

where $A \in \mathbb{Z}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, and $U \in \mathbb{N}$ is an upper bound on the absolute values of the entries in A and $\boldsymbol{b}$.

By [3.2.4] and Corollary 3.2.c, optimal solutions to these problems exist iff there are feasible solutions to the primal and dual problems such that their objective function values are equal, i.e., the polyhedron

$$P = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{n+m} : \boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{b}^T\boldsymbol{y}, A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}, A^T\boldsymbol{y} \leq \boldsymbol{c}\}$$

is nonempty. This provides a connection between LP problem and linear feasibility problem. Using the ellipsoid method, we can determine whether P is nonempty in polynomial time. If so, the method would return a point in a polyhedron perturbed from P, which may not be in P in general. However, it turns out that by carrying out a suitable rounding procedure (which can be completed in polynomial time), a point in P can be found and so an optimal solution to the primal problem can be obtained. This suggests a method for solving such LP in polynomial time. In Section 6, we will explore more methods for solving LP based on the ideas in Section 5.

6 Interior Point Methods

6.0.1 Geometrically speaking, the simplex method solves LP problems by moving between vertices of the feasible region and improving the objective function values at each iteration. The vertices are located at the *boundary* of the feasible region. On the other hand, for interior point methods, we are moving within the *interior* of the feasible region to find optimal solution. Interior point methods serve as viable alternatives to the simplex method, and often outperforms the simplex method for "large" LP problems, due to the theoretical guarantee on the computational efficiency (running in polynomial time).

In Section 6, we will study three common kinds of interior point methods, namely (i) the *affine scaling algorithm*, (ii) the *potential reduction algorithm*, and (iii) the *primal path following algorithm*. Among the three types of interior point methods, the affine scaling algorithm should be the simplest one, so we first study this algorithm.

6.1 Affine Scaling Algorithm

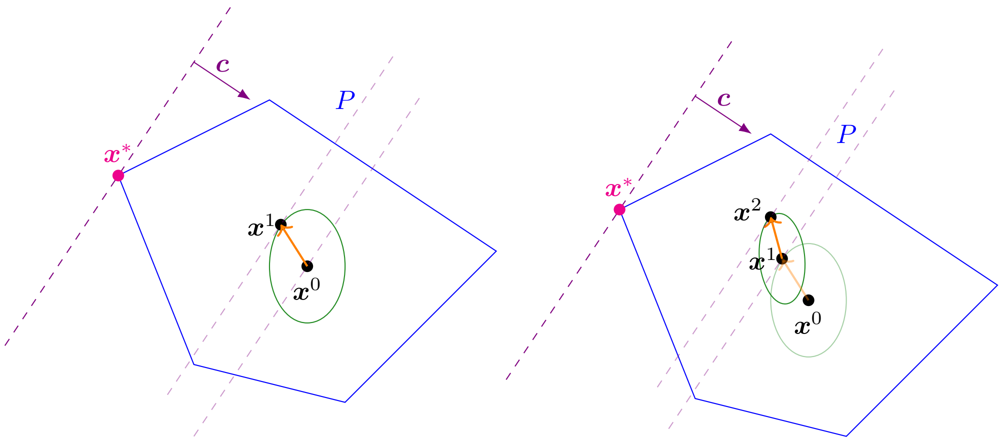
6.1.1 **Geometrical intuition.** Consider a standard form LP

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

and its dual

$$\begin{aligned} \max \quad & \boldsymbol{b}^T\boldsymbol{p} \\ \text{s.t.} \quad & A^T\boldsymbol{p} \leq \boldsymbol{c} \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ denote the feasible region for the primal problem. The **interior** (or **relative interior**) of P is given by $\{\boldsymbol{x} \in P : \boldsymbol{x} > \boldsymbol{0}\}$[8], and its elements are called interior points. Using the property that minimizing the objective function $\boldsymbol{c}^T\boldsymbol{x}$ over an *ellipsoid* is easier than doing that directly over P, the affine scaling algorithm moves between points in the interior of P by performing minimization of $\boldsymbol{c}^T\boldsymbol{x}$ over ellipsoids in the interior, thereby reducing the objective function value at each iteration (like the simplex method). This procedure is graphically represented as follows:



We start with an interior point x_0 of P, and form an ellipsoid S_0 centered at x_0 that is contained in the interior of P. Minimizing the objective function $\boldsymbol{c}^T\boldsymbol{x}$ over the ellipsoid S_0 (easy) yields an optimal

[8] Here, the "interior" is taken with respect to a certain *affine subspace* rather than the Euclidean space $\mathbb{R}^n$.

solution x_1 that is in the interior of P, which has a smaller objective function value than x_0. Next, we again form an ellipsoid centered at x_1 in the interior of P, and the process continues. Intuitively, after sufficiently many iterations, the resulting point would be fairly close to the actual optimal solution $\boldsymbol{x}^*$.

Equipped with some geometrical intuition about the affine scaling algorithm, we will then study its details.

6.1.2 **Forming an ellipsoid in the interior of P.** In the affine scaling algorithm, we would need to form ellipsoids in interior, so let us first investigate how that can be done. To form such ellipsoids, the following result is helpful.

Lemma 6.1.a. Let $\beta \in (0,1)$ be a scalar, $\boldsymbol{y} \in \mathbb{R}^n$ with $\boldsymbol{y} > \boldsymbol{0}$, and let $S := \{\boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^n (x_i - y_i)^2/y_i^2 \le \beta^2\}$. Then we have $\boldsymbol{x} > \boldsymbol{0}$ for all $\boldsymbol{x} \in S$.

Proof. Fix any $\boldsymbol{x} \in S$. For every $i = 1, \dots, n$, we have $(x_i - y_i)^2 \le \beta^2 y_i^2 < y_i^2$, and hence $|x_i - y_i| < y_i$. This particularly implies that $y_i - x_i < y_i$, and so $x_i > 0$. □

To see how this result helps us construct such ellipsoids, consider the following. Fix an interior point $\boldsymbol{y}$ of P and let $Y = \operatorname{diag}(y_1, \dots, y_n)$ denote the $n \times n$ diagonal matrix with (i,i)th entry being y_i for all $i = 1, \dots, n$. As $y_1, \dots, y_n > 0$, Y is invertible, and hence we can express S as

$$S = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^n \frac{(x_i - y_i)^2}{y_i^2} \le \beta^2 \right\} = \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x}-\boldsymbol{y})^T (Y^{-1})^2 (\boldsymbol{x}-\boldsymbol{y}) \le \beta^2\} = \{\boldsymbol{x} \in \mathbb{R}^n : \|Y^{-1}(\boldsymbol{x}-\boldsymbol{y})\| \le \beta\}$$

where $\|\cdot\|$ denotes the Euclidean norm and satisfies that $\|\boldsymbol{v}\|^2 = \boldsymbol{v}^T\boldsymbol{v}$. As $(Y^{-1})^2 = \operatorname{diag}\big(y_1^{-2}, \dots, y_n^{-2}\big)$ is positive definite, S is indeed an ellipsoid centered at $\boldsymbol{y}$. Then, $S_0 := S \cap \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}\}$ would be a section of ellipsoid centered at $\boldsymbol{y}$ that is contained in the interior of P; sometimes we also call S_0 simply as ellipsoid (with respect to P). This provides us a method for obtaining ellipsoids in the interior of P centered at desired points.

6.1.3 **Minimization over ellipsoid.** After forming an ellipsoid S_0 centered at $\boldsymbol{y}$ in the interior of P, we would need to perform minimization over the ellipsoid S_0 in the affine scaling algorithm:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \|Y^{-1}(\boldsymbol{x}-\boldsymbol{y})\| \le \beta \end{aligned}$$

where $\beta \in (0,1)$, $\boldsymbol{y} > \boldsymbol{0}$, and $Y = \operatorname{diag}(y_1, \dots, y_n)$.

Previously we have claimed that solving such minimization problem is "easy". Here we will justify this claim. By "easy", we mean that a closed-form formula is available for getting an optimal solution. Before discussing about it, we first do some preparatory work to simplify the minimization problem a bit.

Let $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{y}$ denote the vector of "movement". By the feasibility of $\boldsymbol{y}$, we have $A\boldsymbol{y} = \boldsymbol{b}$. Also, by construction, every $x \in S_0$ satisfies $A\boldsymbol{x} = \boldsymbol{b}$. These imply that $A\boldsymbol{d} = \boldsymbol{0}$. Now, consider the following problem that performs minimization with respect to $\boldsymbol{d}$ instead of $\boldsymbol{x}$:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{d} \\ \text{s.t.} \quad & A\boldsymbol{d} = \boldsymbol{0} \\ & \|Y^{-1}\boldsymbol{d}\| \le \beta \end{aligned}$$

where $\beta \in (0,1)$ and $Y = \operatorname{diag}(y_1, \dots, y_n)$.

Solving this minimization problem would give us enough ingredient for solving the minimization problem above, since for each optimal solution $\boldsymbol{d}^*$ here, a corresponding optimal solution for the minimization problem above would be $\boldsymbol{x}^* = \boldsymbol{y} + \boldsymbol{d}^*$. Thus we will focus on this (simpler) problem henceforth. The following result provides a closed-form formula for optimal solution to this problem.

Proposition 6.1.b. Suppose that the rows of A are linearly independent and $\boldsymbol{c}$ is not a linear combination of the rows of A. Let $\boldsymbol{y} > \mathbf{0}$. Then an optimal solution to the problem above is

$$\boldsymbol{d}^* = -\beta \frac{Y^2(\boldsymbol{c} - A^T\boldsymbol{p})}{\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\|}$$

with $Y = \operatorname{diag}(y_1, \ldots, y_n)$, $\boldsymbol{p} = (AY^2A^T)^{-1}AY^2\boldsymbol{c}$, and $\beta \in (0,1)$. Furthermore, $\boldsymbol{x}^* = \boldsymbol{y} + \boldsymbol{d}^*$ is an interior point of P and satisfies that $\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{c}^T\boldsymbol{y} - \beta\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\| < \boldsymbol{c}^T\boldsymbol{y}$ *(being in interior and leading to decrease in objective function value after the move).*

Proof. **Showing that $\boldsymbol{d}^*$ and $\boldsymbol{p}$ are well-defined.** First of all, we will show the well-definedness of $\boldsymbol{d}^*$ and $\boldsymbol{p}$ by showing that $\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\| > 0$ and AY^2A^T is invertible respectively. For the former, it follows from the assumption that $\boldsymbol{c}$ is not a linear combination of the rows of A, which implies that $\boldsymbol{c} - A^T\boldsymbol{p} \neq \mathbf{0}$, and hence $Y(\boldsymbol{c} - A^T\boldsymbol{p}) \neq \mathbf{0}$. For the latter, consider the following. Since the rows of A are linearly independent, from linear algebra we know that A and A^T are both invertible. Also, since $\boldsymbol{y} > \mathbf{0}$, we know $y_1^2, \ldots, y_n^2 > 0$, and so Y^2 is invertible. Therefore, AY^2A^T is invertible.

Showing the feasibility of $\boldsymbol{d}^*$. Since $\boldsymbol{y} > 0$, we know Y is invertible, thus

$$Y^{-1}\boldsymbol{d}^* = -\beta \frac{Y(\boldsymbol{c} - A^T\boldsymbol{p})}{\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\|}.$$

Taking $\|\cdot\|$ on both sides then gives $\|Y^{-1}\boldsymbol{d}^*\| = \beta$. So for the feasibility it remains to show that $A\boldsymbol{d}^* = \mathbf{0}$, which is equivalent to $AY^2(\boldsymbol{c} - A^T\boldsymbol{p}) = \mathbf{0}$. This holds because $AY^2(\boldsymbol{c} - A^T\boldsymbol{p}) = AY^2(\boldsymbol{c} - A^T(AY^2A^T)^{-1}AY^2\boldsymbol{c}) = AY^2\boldsymbol{c} - AY^2A^T(AY^2A^T)^{-1}AY^2\boldsymbol{c} = AY^2\boldsymbol{c} - AY^2\boldsymbol{c} = \mathbf{0}$.

Showing the optimality of $\boldsymbol{d}^*$. For every feasible solution $\boldsymbol{d}$ to the problem above, we have $A\boldsymbol{d} = \mathbf{0}$ and $\|Y^{-1}\boldsymbol{d}\| \leq \beta$, and so

$$\begin{aligned}
\boldsymbol{c}^T\boldsymbol{d} &\overset{(A\boldsymbol{d}=\mathbf{0})}{=} (\boldsymbol{c}^T - \boldsymbol{p}^TA)\boldsymbol{d} = (\boldsymbol{c}^T - \boldsymbol{p}^TA)YY^{-1}\boldsymbol{d} \\
&= (Y(\boldsymbol{c} - A^T\boldsymbol{p}))^TY^{-1}\boldsymbol{d} \underset{(\|\boldsymbol{u}\|\|\boldsymbol{v}\| \geq \boldsymbol{u}^T\boldsymbol{v} \geq -\|\boldsymbol{u}\|\|\boldsymbol{v}\|)}{\overset{\text{(Cauchy-Swartz)}}{\geq}} -\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\|\|Y^{-1}\boldsymbol{d}\| \\
&\geq -\beta\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\| \overset{(\boldsymbol{v}^T\boldsymbol{v}=\|\boldsymbol{v}\|^2)}{=} -\beta\frac{[Y(\boldsymbol{c} - A^T\boldsymbol{p})]^T[Y(\boldsymbol{c} - A^T\boldsymbol{p})]}{\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\|} \\
&= -(\boldsymbol{c}^T - \boldsymbol{p}^TA)\beta\frac{Y^2(\boldsymbol{c} - A^T\beta)}{\|\boldsymbol{Y}(\boldsymbol{c} - A^T\boldsymbol{p})\|} = (\boldsymbol{c}^T - \boldsymbol{p}^TA)\boldsymbol{d}^* \overset{(A\boldsymbol{d}^*=\mathbf{0})}{=} \boldsymbol{c}^T\boldsymbol{d}^*,
\end{aligned}$$

which means that $\boldsymbol{d}^*$ is optimal.

Showing that the movement results in an interior point with a drop in objective function value. Note that $\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{c}^T\boldsymbol{y} + \boldsymbol{c}^T\boldsymbol{d}^* = \boldsymbol{c}^T\boldsymbol{y} - \beta\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\|$. Since $\|Y(\boldsymbol{c} - A^T\boldsymbol{p})\| > 0$, we have $\boldsymbol{c}^T\boldsymbol{x}^* < \boldsymbol{c}^T\boldsymbol{y}$. Also, with $A\boldsymbol{d}^* = \mathbf{0}$ and $\|Y^{-1}\boldsymbol{d}^*\| \leq \beta$, we have $A\boldsymbol{x}^* = \boldsymbol{b}$ and $\|Y^{-1}(\boldsymbol{x}^* - \boldsymbol{y})\| \leq \beta$, and the latter implies that $\boldsymbol{x}^* > \mathbf{0}$ by Lemma 6.1.a. Hence, $\boldsymbol{x}^*$ lies in the interior of P. □

Remarks:

- *(Relationship with the iterations of the affine scaling algorithm)* Proposition 6.1.b demonstrates that our geometrical intuition of "forming an ellipsoid and then minimizing over it to move to another point in the interior" actually works. It shows that we can indeed perform such iterations to incrementally decrease the objective function value while remaining in the interior.
- *(Optimal value of the original problem is $-\infty$ if $\boldsymbol{d}^* \geq \mathbf{0}$)* If the optimal solution obtained satisfies $\boldsymbol{d}^* \geq \mathbf{0}$, then we can conclude that the optimal value of the original minimization problem with respect to $\boldsymbol{x}$ is $-\infty$. To see this, note that in such case we would have $\boldsymbol{x}^* + \alpha\boldsymbol{d}^* > \mathbf{0}$ and $A(\boldsymbol{x}^* + \alpha\boldsymbol{d}^*) = A\boldsymbol{x}^* + \alpha A\boldsymbol{d}^* = \boldsymbol{b}$ for all $\alpha > 0$. By Proposition 6.1.b, we know that $\boldsymbol{c}^T\boldsymbol{d}^* = \boldsymbol{c}^T\boldsymbol{x}^* - \boldsymbol{c}^T\boldsymbol{y} < 0$, and hence the objective function value in the original minimization problem can get arbitrarily negative on the feasible region as we increase α.

6.1.4 **Interpreting the vector $\boldsymbol{p}$ as estimate for dual basic solution.** In Proposition 6.1.b, we have a rather complicated expression for the vector $\boldsymbol{p}$, but it actually carries a somewhat intuitive interpretation, and can be viewed as a kind of "estimate" of dual basic solution. To see this, we consider the case where the vector $\boldsymbol{y}$ in Proposition 6.1.b is a nondegenerate basic feasible solution. WLOG, we assume that the first m variables in $\boldsymbol{y}$ are basic, so we can write $\boldsymbol{y} = (y_1, \dots, y_m, 0, \dots, 0)$, and hence $Y = \operatorname{diag}(y_1, \dots, y_m, 0, \dots, 0)$. Let $Y_0 := \operatorname{diag}(y_1, \dots, y_m)$. Then we have $AY = \begin{bmatrix} BY_0 & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{m\times n}$ where $B \in \mathbb{R}^{m\times m}$ is the corresponding basis matrix. Therefore, we can write

$$\begin{aligned}\boldsymbol{p} &= (AY^2A^T)^{-1}AY^2\boldsymbol{c} = (AY(AY)^T)^{-1}AYY\boldsymbol{c} = (BY_0(BY_0)^T)^{-1}BY_0^2\boldsymbol{c}_B\\ &= (B^T)^{-1}(Y_0^{-1})^2B^{-1}BY_0^2\boldsymbol{c}_B = (B^{-1})^T\boldsymbol{c}_B\end{aligned}$$

where $\boldsymbol{c}_B$ is the vector of coefficients in the objective function for the basic variables. Note that the final expression is the same as the associated dual basic solution $\boldsymbol{y}_B$.

But of course, the vector $\boldsymbol{y}$ in Proposition 6.1.b would generally *not* be a nondegenerate basic feasible solution, and so the vector $\boldsymbol{p}$ would *not* be a dual basic solution exactly. However, heuristically, we may treat $\boldsymbol{p}$ as an "estimate" for dual basic solution.

6.1.5 **Optimality condition based on duality gap.** Although the vector $\boldsymbol{p}$ is only an "estimate" for dual basic solution, it can still be used to construct an (exact) optimality condition for the affine scaling algorithm. To start with, note that under the case where $\boldsymbol{y}$ is a nondegenerate basic feasible solution, the vector $\boldsymbol{r} := \boldsymbol{c} - A^T\boldsymbol{p}$ can be expressed as $\boldsymbol{r} = \boldsymbol{c} - A^T(B^{-1})^T\boldsymbol{c}_B$ which is the vector $\bar{\boldsymbol{c}}$ of *reduced costs*. Hence, with a similar idea as before, the vector r can be regarded as an "estimate" for the vector of reduced costs.

Recall from the proof of Lemma 3.3.a that dual feasibility is equivalent to having nonnegative reduced costs. Although the vector $\boldsymbol{r}$ here only serve as an "estimated" reduced cost vector, the same property holds for $\boldsymbol{r}$ *always*, because we have $\boldsymbol{r} \geq \boldsymbol{0} \iff \boldsymbol{c} - A^T\boldsymbol{p} \geq \boldsymbol{0} \iff A^T\boldsymbol{p} \leq \boldsymbol{c}$, where the last inequality corresponds to the dual feasibility.

To derive the *optimality* condition, we would need the notion of **duality gap**, which refers to the difference between primal objective function value and dual objective function value: $\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{b}^T\boldsymbol{p}$, where $\boldsymbol{y}$ and $\boldsymbol{p}$ are primal and dual feasible solutions respectively. By weak duality, it is always nonnegative. Due to the primal feasibility of $\boldsymbol{y}$, we can write $\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{b}^T\boldsymbol{p} = \boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{p}^T\boldsymbol{b} = \boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{p}^TA\boldsymbol{y} = \boldsymbol{r}^T\boldsymbol{y}$, so the duality gap is given by $\boldsymbol{r}^T\boldsymbol{y}$. Also, by Corollary 3.2.c, the duality gap $\boldsymbol{r}^T\boldsymbol{y}$ being zero would imply that $\boldsymbol{y}$ and $\boldsymbol{p}$ are optimal solutions to the primal and the dual respectively. From this result, it is natural to anticipate that under such assumptions, if the duality gap $\boldsymbol{r}^T\boldsymbol{y}$ is "small", then $\boldsymbol{y}$ and $\boldsymbol{p}$ would be "near optimal" solutions. This intuition is formalized by the following result.

Proposition 6.1.c. Let $\boldsymbol{y}$ and $\boldsymbol{p}$ be primal and dual feasible solutions respectively with the duality gap satisfying $\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{b}^T\boldsymbol{p} < \varepsilon$. Then, $\boldsymbol{y}$ and $\boldsymbol{p}$ are **ε-optimal** for the primal and dual respectively, i.e., their objective function values are within the distance of ε from the respective optimal values:

$$\begin{gathered}\boldsymbol{c}^T\boldsymbol{y} \leq \boldsymbol{c}^T\boldsymbol{y} < \boldsymbol{c}^T\boldsymbol{y}^* + \varepsilon,\\ \boldsymbol{b}^T\boldsymbol{p}^* - \varepsilon < \boldsymbol{b}^T\boldsymbol{p} \leq \boldsymbol{b}^T\boldsymbol{p}^*,\end{gathered}$$

where $\boldsymbol{y}^*$ and $\boldsymbol{p}^*$ are optimal solutions for the primal and dual respectively.

Proof. Since $\boldsymbol{y}$ is primal feasible and $\boldsymbol{y}^*$ is primal optimal, we have $\boldsymbol{c}^T\boldsymbol{y}^* \leq \boldsymbol{c}^T\boldsymbol{y}$ by definition. Also, we have $\boldsymbol{b}^T\boldsymbol{p} \leq \boldsymbol{c}^T\boldsymbol{y}$ by weak duality and $\boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{b}^T\boldsymbol{p} < \varepsilon$ by assumption, thus $\boldsymbol{c}^T\boldsymbol{y} < \boldsymbol{b}^T\boldsymbol{p} + \varepsilon \leq \boldsymbol{c}^T\boldsymbol{y} + \varepsilon$. With a similar argument, one can show that $\boldsymbol{b}^T\boldsymbol{p}^* - \varepsilon < \boldsymbol{b}^T\boldsymbol{p} \leq \boldsymbol{b}^T\boldsymbol{p}^*$. □

6.1.6 **Steps in the affine scaling algorithm.** We now have enough ingredients to describe the **affine scaling algorithm** as follows.

(1) *(Inputs)* Specify the following inputs:

i. data of the standard form LP problem: A, $\boldsymbol{b}$, and $\boldsymbol{c}$, where we assume that the rows of A are linearly independent and $\boldsymbol{c}$ is not a linear combination of the rows of A

ii. an initial interior point $\boldsymbol{x}^0$ of the feasible region P

iii. the optimality tolerance $\varepsilon > 0$

iv. parameter $\beta \in (0, 1)$

(2) *(Initialization)* Set $k \leftarrow 0$.

(3) *(Computing estimated dual basic solution and reduced cost)* Based on $\boldsymbol{x}^k = (x_1^k, \ldots, x_n^k)$, set:

- $X_k \leftarrow \operatorname{diag}\big(x_1^k, \ldots, x_n^k\big)$
- $\boldsymbol{p}^k \leftarrow (AX_k^2A^T)^{-1}AX_k^2\boldsymbol{c}$
- $\boldsymbol{r}^k \leftarrow \boldsymbol{c} - A^T\boldsymbol{p}^k$

(4) *(Optimality check)* If $\boldsymbol{r}^k \geq \boldsymbol{0}$ and $(\boldsymbol{x}^k)^T\boldsymbol{r}^k < \varepsilon$, then stop and conclude that $\boldsymbol{x}^k$ and $\boldsymbol{p}^k$ are ε-optimal for the primal and dual respectively.

(5) *(Unboundedness check)* If $-X_k^2\boldsymbol{r}^k \geq \boldsymbol{0}$, then stop and conclude that the optimal value is $-\infty$. [Note: This corresponds to the case where $\boldsymbol{d}^* \geq \boldsymbol{0}$ in Proposition 6.1.b.]

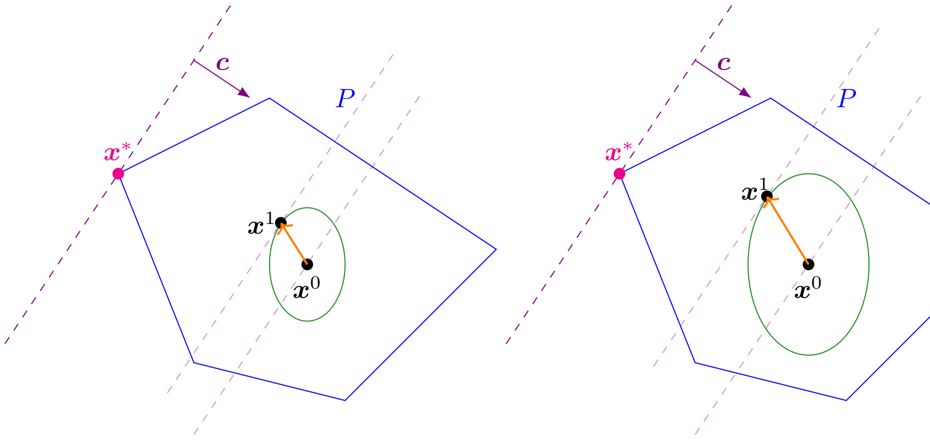
(6) *(Moving to another point in the interior and repeating prior steps)* Set $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k - \beta X_k^2\boldsymbol{r}^k/\|X_k\boldsymbol{r}^k\|$ and then repeat (3)-(5) but with $k \leftarrow k+1$.

[Note: The point $\boldsymbol{x}^{k+1}$ is an optimal solution to the following minimization problem:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \|{X_k}^{-1}(\boldsymbol{x} - \boldsymbol{x}^k)\| \leq \beta \end{aligned}$$

Here the feasible region is a section of ellipsoid centered at $\boldsymbol{x}^k$ that is contained in the interior of P.]

6.1.7 **Step size for the affine scaling algorithm.** For computational efficiency, an important quantity to be determined for the affine scaling algorithm is its step size. In each iteration, we move to from an interior point to another by the formula $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \beta X_k^2\boldsymbol{r}^k/\|X_k\boldsymbol{r}^k\|$. One clear factor that influences the size of the movement (*step size*) is the value of β. The larger the β (while being between 0 and 1), the larger the step size; larger step size signifies more efficient progress and thus is often preferred. Geometrically, this phenomenon occurs because increasing the value of β would enlarge the ellipsoids formed for the iterations, and hence further movements can be obtained:



Smaller β Larger β

Another factor that influences the step size is the way we choose to move between interior points. From the pictures above, it appears that we can indeed move *beyond* the ellipsoid without exiting the interior, which can result in an even larger step size. But to do that, we would need to change the way we move between interior points, which leads to the discussion of *long-step variants* of the affine scaling algorithm.

6.1.8 **Long-step variants.** When we choose to move between interior points according to $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \beta X_k^2 \boldsymbol{r}^k / \|X_k \boldsymbol{r}^k\|$ in the affine scaling algorithm, this version of algorithm is said to be **short-step**; as we shall see, the step sizes for this kind of movement would be relatively small, hence the name "short-step". To get long-step variants, we modify the formula above by replacing the Euclidean norm $\|\cdot\|$ by another function that gives smaller values. For every $\boldsymbol{u} \in \mathbb{R}^n$, the **maximum norm** (or **∞-norm**) of $\boldsymbol{u}$ is $\|\boldsymbol{u}\|_\infty = \max\{|u_i| : i = 1, \dots, n\}$. It satisfies the property that $\|\boldsymbol{u}\|_\infty \le \|\boldsymbol{u}\|$ since we have $|u_i| = \sqrt{u_i^2} \le \sqrt{u_1^2 + \cdots + u_n^2} = \|\boldsymbol{u}\|$ for all $i = 1, \dots, n$. So, by changing the Euclidean norm $\|\cdot\|$ to the maximum norm $\|\cdot\|_\infty$ in the formula, we can have longer steps. The only thing left to show is that the formula would still lead to a movement to an *interior point* and would not "overshoot". This is assured by the following result.

Proposition 6.1.d. Let $\boldsymbol{x}^k$ be an interior point of the feasible region P, and let X_k, $\boldsymbol{r}^k$, and β be as defined in the affine scaling algorithm. Then $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \beta X_k^2 \boldsymbol{r}^k / \|X_k \boldsymbol{r}^k\|_\infty$ is an interior point of P.

Proof. Since $X_k^{-1}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = -\beta X_k^2 \boldsymbol{r}^k / \|X_k \boldsymbol{r}^k\|_\infty$, we have

$$\max\left\{ \frac{|x_i^{k+1} - x_i^k|}{x_i^k} : i = 1, \dots, n \right\} = \|X_k^{-1}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)\|_\infty = \beta \frac{\|X_k \boldsymbol{r}^k\|_\infty}{\|X_k \boldsymbol{r}^k\|_\infty} = \beta.$$

So, for all $i = 1, \dots, n$, we have $|x_i^{k+1} - x_i^k| / x_i^k \le \beta < 1$, which implies that $x_i^k - x_i^{k+1} < x_i^k$, and hence $x_i^{k+1} > 0$.

Furthermore, similar to the feasibility part of the proof of Proposition 6.1.b, we have $AX_k^2(\boldsymbol{c} - A^T \boldsymbol{p}^k) = AX_k^2(\boldsymbol{c} - A^T (AX_k^2 A^T)^{-1} AX_k^2 \boldsymbol{c}) = AX_k^2 \boldsymbol{c} - AX_k^2 A^T (AX_k^2 A^T)^{-1} AX_k \boldsymbol{c} = \boldsymbol{0}$, which implies that $A\boldsymbol{x}^{k+1} = A\boldsymbol{x}^k + \boldsymbol{0} = \boldsymbol{b}$. Hence, $\boldsymbol{x}^{k+1}$ is an interior point of P. □

Using the maximum norm instead of the Euclidean norm is a long-step variant, but it is not the only one. Another more popular long-step variant is to replace the Euclidean norm $\|\cdot\|$ by the function γ given by $\gamma(\boldsymbol{u}) = \max\{u_i : u_i > 0\}$ for all $\boldsymbol{u} \in \mathbb{R}^n$. Noting that $\gamma(\boldsymbol{u}) = \max\{u_i : u_i > 0\} \le \max\{|u_i| : i = 1, \dots, n\} = \|\boldsymbol{u}\|_\infty$, this long-step variant can lead to a larger step size, explaining why it is more popular. We can also establish a result analogous to Proposition 6.1.d for this long-step variant using a similar argument.

6.1.9 **Convergence.** Based on our intuitive geometrical understanding, the affine scaling algorithm seems to allow us to get "very close" to the optimal solution after many iterations, since at each iteration we should be "closer" to the optimal solution. This intuition is theoretically justified by the following result, under some regularity assumptions.

Theorem 6.1.e. Assume that:

(a) The matrix A has linearly independent rows.

(b) The vector $\boldsymbol{c}$ is not a linear combination of the rows of A.

(c) There exists an optimal solution.

(d) There exists an interior point of the feasible region.

If we apply the long-step affine scaling algorithm with $\varepsilon = 0$, the function γ, and $0 < \beta < 2/3$, then the sequences $\{\boldsymbol{x}^k\}$ and $\{\boldsymbol{p}^k\}$ converge to some primal and dual optimal solutions respectively.

[Note: More precisely, setting "$\varepsilon = 0$" means that for the optimality check in [6.1.6], we require $(\boldsymbol{x}^k)\boldsymbol{r}^k = 0$ *(exact optimality)*.]

Proof. Omitted. □

6.1.10 **Initialization.** Apart from the theoretical guarantee on the convergence, another important consideration in the implementation of affine scaling algorithm is *initialization*. Recall from [6.1.6] that we need to start with an initial interior point $\boldsymbol{x}^0$ of the feasible region in the affine scaling algorithm. Previously, for the simplex method, one way to handle the initialization is to use the big-M method ([2.3.10]). Here, we apply a similar idea to find an initial interior point.

Our original problem is the usual standard form LP problem:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

Then, consider the following LP problem that is modified from the original, with a new variable x_{n+1} added:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} + Mx_{n+1} \\ \text{s.t.} \quad & A\boldsymbol{x} + (\boldsymbol{b} - A\boldsymbol{1})x_{n+1} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, x_{n+1} \geq 0, \end{aligned}$$

where $M > 0$ is a "large" constant. For this modified problem, we can see that $(\boldsymbol{x}, x_{n+1}) = (\boldsymbol{1}, 1) > \boldsymbol{0}$ is an interior point of the feasible region, since we have $A\boldsymbol{1} + (\boldsymbol{b} - A\boldsymbol{1})(1) = \boldsymbol{b}$.

Like the big-M method, as long as the original problem has an optimal solution and M is sufficiently large, the optimal solution to this modified problem would always satisfy $x_{n+1} = 0$, yielding an optimal solution to the original problem.

Proposition 6.1.f. If the original problem has an optimal solution, then every optimal solution of the modified problem satisfies $x_{n+1} = 0$, provided that M is sufficiently large.

Proof. Omitted. □

6.1.11 **Computational performance.** An important practical aspect about the affine scaling algorithm is its *computational performance*. While the affine scaling algorithm is observed to have good performance in practice, whether the affine scaling algorithm runs in polynomial time remains an open question. However, we can make the following observations:

(a) *(Behaviour when initializing near a vertex)* When the affine scaling algorithm is initialized near a vertex, it has a similar behaviour as the simplex method in the sense that it also travels near the "edge" of the feasible region. In view of this, the answer to the open question may be more likely to be negative, as we know that the simplex method does not run in polynomial time.

(b) *(More progress when being far from boundary)* It is also observed that the affine scaling algorithm moves in larger steps and makes more progress when the position is far from the boundary of the feasible region (lying "deep" inside the feasible region). This can be explained by noting that larger ellipsoids can be formed when the center is far from boundary, thereby resulting in larger steps.

6.2 Potential Reduction Algorithm

6.2.1 **Idea.** Inspired by the observation in [6.1.11]b, we would like to design an algorithm that allow the iterates to get closer to the optimal solution, while making them far from the boundary, so that more progress can be made in the movements generally.

More specifically, consider the following standard form LP

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

and its dual (with a vector $\boldsymbol{s}$ of slack variables added)

$$\begin{aligned} \max \quad & \boldsymbol{b}^T\boldsymbol{p} \\ \text{s.t.} \quad & A^T\boldsymbol{p} + \boldsymbol{s} = \boldsymbol{c} \\ & \boldsymbol{s} \geq \boldsymbol{0} \end{aligned}$$

Then, we want to achieve the following two goals in the algorithm:

(a) Minimize the objective function value $\boldsymbol{c}^T\boldsymbol{x}$ in the standard form LP problem.

(b) Keep the iterates far from boundary.

However, these two goals are indeed conflicting, since being away from boundary generally limits the extent of the reduction in the objective function value. Nevertheless, these two goals can still be incorporated simultaneously using the following **potential function**:

$$G(\boldsymbol{x},\boldsymbol{s}) = q\ln(\boldsymbol{s}^T\boldsymbol{x}) - \sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln s_j$$

where $q > n$ is a constant. This leads to the *potential reduction algorithm*, which reduces the potential function during the iterations. Through minimizing the potential function $G(\boldsymbol{x},\boldsymbol{s})$ rather than the original objective function $\boldsymbol{c}^T\boldsymbol{x}$, the iterates would still move towards an optimal solution, but along a trajectory that is relatively far from the boundary.

6.2.2 **Interpretation of potential function.** To better understand the potential function, we can consider the following two parts separately:

- $q\ln(\boldsymbol{s}^T\boldsymbol{x})$: This is related to the duality gap, by noting that

$$\boldsymbol{s}^T\boldsymbol{x} = (\boldsymbol{c} - A^T\boldsymbol{p})^T\boldsymbol{x} = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{p}^T\underbrace{\boldsymbol{A}\boldsymbol{x}}_{\boldsymbol{b}} = \underbrace{\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{p}}_{\text{duality gap}}$$

  for all primal and dual feasible solutions $\boldsymbol{x}$ and $(\boldsymbol{p},\boldsymbol{s})$ respectively.
- $-\sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln s_j$: This expression would become very positive if the primal or dual solution gets close to the boundary, thereby *penalizing* the closedness to the boundary.

Therefore, the potential function comprises of a term that capture the closedness to optimality (through the duality gap), and a penalty term for the closedness to the boundary. The logarithms in the penalty term amplify the penalties when the solutions are *extremely* close to the boundary, and correspondingly the logarithm is added to the first term for balancing the effects from the two sources.

6.2.3 **A lower bound on the potential function.** Now, we start discussing how the potential function can help us to find an optimal solution. The following preliminary result provides a lower bound on the potential function, which is utilized for justifying the potential reduction theorem.

Lemma 6.2.a. For all $\boldsymbol{x} > \boldsymbol{0}$ and $\boldsymbol{s} > \boldsymbol{0}$, we have

$$G(\boldsymbol{x},\boldsymbol{s}) \geq n\ln n + (q-n)\ln(\boldsymbol{s}^T\boldsymbol{x})$$

Proof. We first write

$$G(\boldsymbol{x},\boldsymbol{s}) = q\ln(\boldsymbol{s}^T\boldsymbol{x}) - \sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln s_j = (q-n)\ln(\boldsymbol{s}^T\boldsymbol{x}) + n\ln(\boldsymbol{s}^T\boldsymbol{x}) - \sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln s_j.$$

It then remains to show that $n\ln(\boldsymbol{s}^T\boldsymbol{x}) - \sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln s_j \geq n\ln n$. To show this, we use the AM-GM inequality as follows:

$$\begin{aligned}
n\ln(\boldsymbol{s}^T\boldsymbol{x}) - \sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln s_j &= n\ln(\boldsymbol{s}^T\boldsymbol{x}) - \ln\left(\prod_{j=1}^{n}x_j\right) - \ln\left(\prod_{j=1}^{n}s_j\right) \\
&= \ln\underbrace{\left(\sum_{j=1}^{n}x_js_j\right)^n}_{\underset{\text{(AM-GM)}}{\geq} n^n\prod_{j=1}^{n}x_js_j} - \ln\left(\prod_{j=1}^{n}x_js_j\right) \\
&\geq \ln\left(n^n\prod_{j=1}^{n}x_js_j/\prod_{j=1}^{n}x_js_j\right) = \ln n^n = n\ln n.
\end{aligned}$$

□

[Note: This result informs us that when the potential function $G(\boldsymbol{x}, \boldsymbol{s})$ takes small values, the duality gap $\boldsymbol{s}^T\boldsymbol{x}$ would also be small. This gives us some idea on why minimizing the potential function can lead to optimal solution still.]

6.2.4 **Getting ε-optimal solutions by reducing potential function iteratively.** The following result provides a theoretical guarantee that, as long as we can reduce the potential function by a sufficiently large amount in each iteration, ε-optimal solutions for both the primal and dual problems can be obtained after finitely many iterations; indeed, the result below provides an explicit formula for the upper bound on the number of iterations needed. It serves as the foundation for the potential reduction algorithm.

Theorem 6.2.b. Let $\boldsymbol{x}^0$ and $(\boldsymbol{p}^0, \boldsymbol{s}^0)$ be primal and dual feasible solutions respectively, with $\boldsymbol{x}^0 > \boldsymbol{0}$ and $\boldsymbol{s}^0 > \boldsymbol{0}$. Let $\varepsilon > 0$ be the optimal tolerance. Then, every algorithm that (i) maintains both the primal and dual feasibility and (ii) reduces the potential function $G(\boldsymbol{x}, \boldsymbol{s})$ by at least a fixed constant $\delta > 0$, in each iteration, results in solutions $\boldsymbol{x}^K$ and $(\boldsymbol{p}^K, \boldsymbol{s}^K)$ for both the primal and dual problems having the duality gap $(\boldsymbol{s}^K)^T\boldsymbol{x}^K = \boldsymbol{c}^T\boldsymbol{x}^K - \boldsymbol{b}^T\boldsymbol{p}^K \le \varepsilon$[9] after

$$K = \left\lceil \frac{G(\boldsymbol{x}^0, \boldsymbol{s}^0) + (q-n)\ln(1/\varepsilon) - n\ln n}{\delta} \right\rceil$$

iterations.

Proof. By assumption the algorithm has the property that $G(\boldsymbol{x}^{k+1}, \boldsymbol{s}^{k+1}) - G(\boldsymbol{x}^k, \boldsymbol{s}^k) \le -\delta$ for every $k = 0, 1, 2, \dots$. Hence after K iterations we have $G(\boldsymbol{x}^K, \boldsymbol{s}^K) \le G(\boldsymbol{x}^0, \boldsymbol{s}^0) - K\delta$. For the K specified above, we then get

$$n\ln n + \underbrace{(q-n)}_{>0}\ln((\boldsymbol{s}^K)^T\boldsymbol{x}^K) \overset{\text{(Lemma 6.2.a)}}{\le} G(\boldsymbol{x}^K, \boldsymbol{s}^K) \le -(q-n)\ln(1/\varepsilon) + n\ln n = (q-n)\ln\varepsilon + n\ln n,$$

which implies that $(\boldsymbol{s}^K)^T\boldsymbol{x}^K \le \varepsilon$. □

6.2.5 **Idea of the potential reduction algorithm.** Theorem 6.2.b serves a theoretical justification on why algorithms based on reducing the potential function work. Next, we are going to answer an important related question: *How* can such algorithm be designed? In the following, we will explore the intuitive idea behind the potential reduction algorithm.

In each iteration of the affine scaling algorithm, recall that we are solving the following minimization problem:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{d} = \boldsymbol{c}^T(\boldsymbol{x}+\boldsymbol{d}) - \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{d} = \boldsymbol{0} \\ & \|X^{-1}\boldsymbol{d}\| \le \beta \end{aligned}$$

where $\beta \in (0,1)$ and $X = \operatorname{diag}(x_1, \dots, x_n)$, with $\boldsymbol{x} = (x_1, \dots, x_n) > \boldsymbol{0}$ being the primal feasible solution in the iteration; here the objective function is the change in the objective value after the movement.

To adapt it for the case here, it is natural to consider the following problem:

$$\begin{aligned} \min \quad & G(\boldsymbol{x}+\boldsymbol{d}, \boldsymbol{s}) - G(\boldsymbol{x}, \boldsymbol{s}) \\ \text{s.t.} \quad & A\boldsymbol{d} = \boldsymbol{0} \\ & \|X^{-1}\boldsymbol{d}\| \le \beta \end{aligned}$$

where the objective function is the change in the potential function after the movement. However, such objective would generally be nonlinear and so the optimization problem is usually hard to solve.

[9] Previously, the ε-optimality requires "$< \varepsilon$", but here we only have "$\le \varepsilon$". Nevertheless, we usually do not care much about this subtlety, and still regard such solutions as "ε-optimal".

To make the problem more manageable, we approximate the objective function $G(\boldsymbol{x}+\boldsymbol{d},\boldsymbol{s})-G(\boldsymbol{x},\boldsymbol{s})$ through a first-order Taylor expansion of G at $(\boldsymbol{x},\boldsymbol{s})$: $G(\boldsymbol{x}+\boldsymbol{d},\boldsymbol{s})-G(\boldsymbol{x},\boldsymbol{s})=\nabla_{\boldsymbol{x}}G(\boldsymbol{x},\boldsymbol{s})^T\boldsymbol{d}+O(\|\boldsymbol{d}\|^2)$, where $\nabla_{\boldsymbol{x}}$ denotes the gradient containing partial derivatives with respect to the variables in $\boldsymbol{x}$.

After performing such approximation, we consider the following simplified problem:

$$\begin{aligned} \min\quad & \nabla_{\boldsymbol{x}}G(\boldsymbol{x},\boldsymbol{s})^T\boldsymbol{d} \\ \text{s.t.}\quad & A\boldsymbol{d}=\boldsymbol{0} \\ & \|X^{-1}\boldsymbol{d}\|\leq\beta \end{aligned}$$

One important observation is that this problem is essentially the problem handled in [6.1.3], with "$\boldsymbol{c}$" being $\widehat{\boldsymbol{c}}:=\nabla_{\boldsymbol{x}}G(\boldsymbol{x},\boldsymbol{s})$, and therefore admits a closed-form solution as suggested by Proposition 6.1.b, namely

$$\boxed{\boldsymbol{d}^*=-\beta\frac{X^2(\widehat{\boldsymbol{c}}-A^T\widehat{\boldsymbol{p}})}{\|X(\widehat{\boldsymbol{c}}-A^T\widehat{\boldsymbol{p}})\|}=-\beta X\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}}$$

with $\widehat{\boldsymbol{p}}=(AX^2A^T)^{-1}AX^2\widehat{\boldsymbol{c}}$ and $\boldsymbol{u}=X(\widehat{\boldsymbol{c}}-A^T\widehat{\boldsymbol{p}})$.

6.2.6 **Further expressions of $\boldsymbol{u}$ and the change $G(\boldsymbol{x}+\boldsymbol{d}^*,\boldsymbol{s})-G(\boldsymbol{x},\boldsymbol{s})$.** Since we can explicitly derive the expression of $\nabla_{\boldsymbol{x}}G(\boldsymbol{x},\boldsymbol{s})$, we are able to obtain a more explicit expression of $\boldsymbol{u}$. Noting that

$$\widehat{c}_i=\frac{\partial G(\boldsymbol{x},\boldsymbol{s})}{\partial x_i}=\frac{\partial}{\partial x_i}\left(q\ln\sum_{i=1}^n s_ix_i-\sum_{i=1}^n\ln x_i-\sum_{i=1}^n\ln s_i\right)=\frac{qs_i}{\boldsymbol{s}^T\boldsymbol{x}}-\frac{1}{x_i}\quad\text{for all } i=1,\dots,n,$$

we get $(X\widehat{\boldsymbol{c}})_i=x_i\widehat{\boldsymbol{c}}_i=\frac{q}{\boldsymbol{s}^T\boldsymbol{x}}x_is_i-1$, and hence $X\widehat{\boldsymbol{c}}=\frac{q}{\boldsymbol{s}^T\boldsymbol{x}}X\boldsymbol{s}-\boldsymbol{1}$ with $\boldsymbol{1}=(1,\dots,1)\in\mathbb{R}^n$. It follows that

$$\boldsymbol{u}=(I-XA^T(AX^2A^T)^{-1}AX)X\widehat{\boldsymbol{c}}=\boxed{(I-(AX)^T(AX^2A^T)^{-1}AX)\left(\frac{q}{\boldsymbol{s}^T\boldsymbol{x}}X\boldsymbol{s}-\boldsymbol{1}\right)}.$$

For the change $G(\boldsymbol{x}+\boldsymbol{d}^*,\boldsymbol{s})-G(\boldsymbol{x},\boldsymbol{s})$ in potential function, we have

$$G(\boldsymbol{x}+\boldsymbol{d}^*,\boldsymbol{s})-G(\boldsymbol{x},\boldsymbol{s})=\nabla_{\boldsymbol{x}}G(\boldsymbol{x},\boldsymbol{s})^T\boldsymbol{d}^*+O(\|\boldsymbol{d}^*\|^2)\overset{\text{(Proposition 6.1.b)}}{=}-\beta\|\boldsymbol{u}\|+O(\beta^2).$$

It can be shown that with a fixed $\gamma\in(0,1)$, if $\|\boldsymbol{u}\|\geq\gamma$, then the potential function is guaranteed to be reduced by at least a constant (depending on the value of γ) in each iteration, and therefore Theorem 6.2.b is applicable. In case $\|\boldsymbol{u}\|<\gamma$, updating the primal feasible solution alone would not be enough for achieving a sufficient reduction in the potential function, and hence one would need to update the *dual* feasible solution also to achieve that. This leads to the *potential reduction algorithm*.

6.2.7 **Steps in the potential reduction algorithm.** The potential reduction algorithm is as follows.

(1) *(Inputs)* Specify the following inputs:

  i. data of the standard form LP problem: A, $\boldsymbol{b}$, and $\boldsymbol{c}$, where A is assumed to have linearly independent rows
  ii. the initial primal and dual feasible solutions: $\boldsymbol{x}^0$ and $(\boldsymbol{p}^0,\boldsymbol{s}^0)$, with $\boldsymbol{x}^0>\boldsymbol{0}$ and $\boldsymbol{s}^0>\boldsymbol{0}$
  iii. the optimality tolerance $\varepsilon>0$
  iv. the parameters $\beta\in(0,1)$, $\gamma\in(0,1)$, and $q>n$

(2) *(Initialization)* Set $k\leftarrow 0$.

(3) *(Optimality check)* If $(\boldsymbol{s}^k)^T\boldsymbol{x}^k<\varepsilon$, then stop and conclude that $\boldsymbol{x}^k$ and $\boldsymbol{p}^k$ are ε-optimal for the primal and dual respectively.

(4) *(Computing update direction)* Based on $\boldsymbol{x}^k=(x_1^k,\dots,x_n^k)$, set:

- $X_k\leftarrow\operatorname{diag}(x_1^k,\dots,x_n^k)$
- $\bar{A}^k\leftarrow(AX_k)^T(AX_k^2A^T)^{-1}AX_k$

- $\boldsymbol{u}^k \leftarrow (I - \bar{A}^k)\left(\frac{q}{(\boldsymbol{s}^k)^T\boldsymbol{x}^k} X_k\boldsymbol{s}^k - \mathbf{1}\right)$
- $\boldsymbol{d}^k \leftarrow -\beta X_k\boldsymbol{u}^k/\|\boldsymbol{u}^k\|$

(5) *(Primal step)* If $\|\boldsymbol{u}^k\| \geq \gamma$, then set:

- $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \boldsymbol{d}^k$
- $\boldsymbol{s}^{k+1} \leftarrow \boldsymbol{s}^k$
- $\boldsymbol{p}^{k+1} \leftarrow \boldsymbol{p}^k$

(6) *(Dual step)* If $\|\boldsymbol{u}^k\| < \gamma$, then set:

- $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k$
- $\boldsymbol{s}^{k+1} \leftarrow \frac{(\boldsymbol{s}^k)^T\boldsymbol{x}^k}{q}(X_k)^{-1}(\boldsymbol{u}^k + \mathbf{1})$
- $\boldsymbol{p}^{k+1} \leftarrow \boldsymbol{p}^k + (AX_k^2A^T)^{-1}(AX_k)\left(X_k\boldsymbol{s}^k - \frac{(\boldsymbol{s}^k)^T\boldsymbol{x}^k}{q}\mathbf{1}\right)$.

(7) *(Repeating prior steps)* Set $k \leftarrow k+1$ and go back to (3).

[Note: Here we shall omit the details about the derivation of the formulas in the dual step.]

6.3 Primal Path Following Algorithm

6.3.1 The last type of interior point methods to be discussed in Section 6 is known as the *primal path following algorithm*. Although it is perhaps the most complex one among the three here, it has the most superior performance and has wide applicability in practice. As suggested by its name, the algorithm is about following a path in the primal feasible region, in order to find an optimal solution. Indeed, this path following algorithm is somewhat similar to the potential reduction algorithm conceptually, as both take the closedness to the boundary into the account and try to travel along a trajectory (path) that is relatively far from the boundary, through introducing a penalty term.

6.3.2 **Barrier problems.** We shall motivate the path following algorithm by the notion of *barrier problems*. Consider a standard form LP problem

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \mathbf{0} \end{aligned}$$

In solving this LP problem, a major difficulty arises from the inequality constraint $\boldsymbol{x} \geq \mathbf{0}$; generally, equality constraints are easier to deal with. One method to "get rid of" the inequality constraint (more precisely, "converting" it to some other form) is to analyze the associated *barrier problems* instead, as specified in the following.

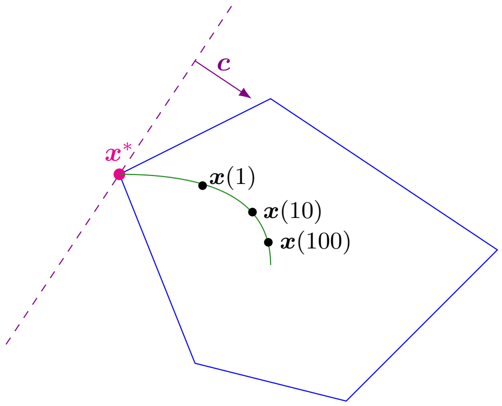
Let $\mu > 0$. The function $B_\mu(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} \overbrace{-\mu\sum_{j=1}^{n}\ln x_j}^{\text{penalty term}}$ is known as the **(log) barrier function**, where we conventionally define $B_\mu(\boldsymbol{x}) := \infty$ if $x_j \leq 0$ for some j. Then, the family of **barrier problems** is given by

$$\begin{aligned} \min \quad & B_\mu(\boldsymbol{x}) \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

for every $\mu > 0$; each $\mu > 0$ corresponds to one barrier problem, and the larger the μ is, the stronger the penalty is in effect. Here, we assume that the parameters are set such that the barrier problem has a unique optimal solution $\boldsymbol{x}^* = \boldsymbol{x}(\mu)$ for every $\mu > 0$. Note that the optimal solution must satisfy that $\boldsymbol{x}(\mu) > \mathbf{0}$, for otherwise the objective function of the barrier problem would be $-\infty$ at the optimal solution $\boldsymbol{x}(\mu)$, which is impossible. Particularly, this implies that $\boldsymbol{x}(\mu)$ always lies in the original feasible region P for all $\mu > 0$.

6.3.3 **Central path and analytic center.** Considering $\boldsymbol{x}(\mu)$ as a function of μ with $\boldsymbol{x} : (0, \infty) \to P$, varying the parameter μ would produce a *path* traced by $\boldsymbol{x}(\mu)$ inside the feasible region P, which is known as the **central path**. It can be shown that $\lim_{\mu \to 0} \boldsymbol{x}(\mu)$ exists and equals an optimal solution to the original LP problem; intuitively, this happens since with a very small μ, the penalty would be negligible and hence the optimal solution to the barrier problem would be extremely close to the optimal solution to the original LP problem.

Further insights about the central path can be gained by inspecting the following graphical illustration:



As we increase μ, the penalty effect is more significant, creating a stronger force to "push away" the resulting optimal solution $\boldsymbol{x}(\mu)$ from the boundary of the feasible region, which makes $\boldsymbol{x}(\mu)$ farther from the actual optimal solution $\boldsymbol{x}^*$, but also being more "deep inside" the feasible region. Intuitively, letting $\mu \to \infty$ would make $\boldsymbol{x}(\mu)$ approach to the "center" of the feasible region (a point that is the "farthest" from the boundary). Such point $\lim_{\mu \to \infty} \boldsymbol{x}(\mu)$ is said to be the **analytic center** of the polyhedron P. [Note: There are multiple kinds of centers for a polyhedron, just like the case for triangle, which has different types of "center" like centroid, incenter, circumcenter, and orthocenter.]

The analytic center is also the optimal solution to the "penalty-only" problem

$$\begin{aligned} \min \quad & -\sum_{j=1}^{n} \ln x_j \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}. \end{aligned}$$

Intuitively, this is because as $\mu \to \infty$, the penalty effect dominates and the original objective function would not play a role in affecting the resulting optimal solution anymore.

6.3.4 **Solving barrier problems with approximations.** As suggested by [6.3.3], one way to obtain/approximate an optimal solution to the original LP problem is to compute $\boldsymbol{x}(\mu)$ with μ being close to zero. However, barrier problem is generally *difficult* to solve ⚠ due to the nonlinearity of objective function. Therefore, like the potential reduction algorithm, we need to first simplify the barrier problem through some approximations.

In the case of potential reduction algorithm, we need to simplify a problem with feasible region being an ellipsoid *(nonlinear)* and we have used a first-order Taylor approximation. On the other hand, here the feasible region of optimization problem is only a polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}\}$ *(linear)*, and only using a first-order Taylor approximation here would not be too interesting (and also not enough for the theoretical results to work!). In view of this, we will consider a *second-order* Taylor approximation here.

Fix any $\boldsymbol{x} > \mathbf{0}$. Note that

$$\begin{aligned}
\frac{\partial B_\mu(\boldsymbol{x})}{\partial x_i} &= c_i - \frac{\mu}{x_i} \quad \forall i = 1, \ldots, n, \\
\frac{\partial^2 B_\mu(\boldsymbol{x})}{\partial x_i^2} &= \frac{\mu}{x_i^2} \quad \forall i = 1, \ldots, n, \\
\frac{\partial^2 B_\mu(\boldsymbol{x})}{\partial x_i \, \partial x_j} &= 0 \quad \forall i \neq j.
\end{aligned}$$

Hence, with *sufficiently small* $\boldsymbol{d}$, by second-order Taylor expansion we have

$$B_\mu(\boldsymbol{x} + \boldsymbol{d}) - B_\mu(\boldsymbol{x}) \approx \frac{\partial B_\mu(\boldsymbol{x})}{\partial x_i} d_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 B_\mu(\boldsymbol{x})}{\partial x_i \, \partial x_j} d_i d_j = (\boldsymbol{c}^T - \mu \mathbf{1}^T X^{-1})\boldsymbol{d} + \frac{1}{2} \mu \boldsymbol{d}^T (X^{-1})^2 \boldsymbol{d}$$

where $X = \operatorname{diag}(x_1, \ldots, x_n)$. Based on this approximation, we will then consider the following minimization problem, in terms of direction $\boldsymbol{d}$ at $\boldsymbol{x}$:

$$\begin{aligned}
\min \quad & (\boldsymbol{c}^T - \mu \mathbf{1}^T X^{-1})\boldsymbol{d} + \frac{1}{2} \mu \boldsymbol{d}^T (X^{-1})^2 \boldsymbol{d} \\
\text{s.t.} \quad & A\boldsymbol{d} = \mathbf{0}.
\end{aligned}$$

6.3.5 **Newton direction.** Like the case of potential reduction algorithm, this problem admits closed-form solution, but this time it does not directly follow from previous results; we need to do some work on solving the problem. More specifically, we will utilize the method of *Lagrange multiplier* (from MATH2211). First, associate a vector $\boldsymbol{p}$ of Lagrange multipliers to the equality constraint $A\boldsymbol{d} = \mathbf{0}$ and form the Lagrangian function

$$L(\boldsymbol{d}, \boldsymbol{p}) = \underbrace{(\boldsymbol{c}^T - \mu \mathbf{1}^T X^{-1})\boldsymbol{d} + \frac{1}{2} \mu \boldsymbol{d}^T (X^{-1})^2 \boldsymbol{d}}_{\text{objective}} - \boldsymbol{p}^T \overbrace{A\boldsymbol{d}}^{\text{LHS of equality constraint}}.$$

Then, we need to solve the following system (with $m + n$ equations and $m + n$ unknowns):

$$\begin{aligned}
\frac{\partial L(\boldsymbol{d}, \boldsymbol{p})}{\partial d_j} &= 0 \quad \forall j = 1, \ldots, n, \\
\frac{\partial L(\boldsymbol{d}, \boldsymbol{p})}{\partial p_i} &= 0 \quad \forall i = 1, \ldots, m.
\end{aligned}$$

More explicitly, it can be expressed in vector form as

$$\begin{aligned}
\boldsymbol{c} - \mu X^{-1} \mathbf{1} + \mu (X^{-1})^2 \boldsymbol{d} - A^T \boldsymbol{p} &= \mathbf{0}, \\
A\boldsymbol{d} &= \mathbf{0}.
\end{aligned}$$

Solving this system gives the solutions

$$\begin{aligned}
\boldsymbol{d}(\mu) &= \left(I - X^2 A^T (A X^2 A^T)^{-1} A\right) \left(X \mathbf{1} - \frac{1}{\mu} X^2 \boldsymbol{c}\right), \\
\boldsymbol{p}(\mu) &= (A X^2 A^T)^{-1} A (X^2 \boldsymbol{c} - \mu X \mathbf{1}).
\end{aligned}$$

The solution $\boldsymbol{d}(\mu)$ is called the **Newton direction**. We have the name "Newton direction" since the method described here is indeed an instance of the *Newton's method* in optimization, which is to optimize recursively *local* quadratic models based on gradient and Hessian. Here, the second-order Taylor expansion involves gradient and Hessian, and also such approximation only works *locally* (with sufficiently small $\boldsymbol{d}$). The Newton direction serves as an useful tool for avoiding the exact computation of $\boldsymbol{x}(\mu)$ in the path following algorithm.

6.3.6 **Approximation of central path.** The idea of using the Newton direction for approximating the central path ("following that path" approximately) is as follows.

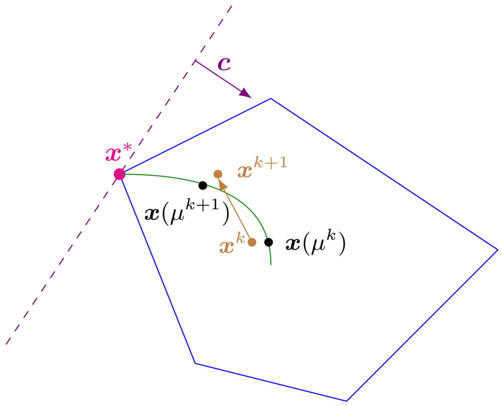
(1) *(Initialization)* Specify $\boldsymbol{x}^0 > 0$, μ^0, and $0 < \alpha < 1$, and set $k \leftarrow 0$.

(2) *(Computing Newton direction)* Using the formula in [6.3.5], compute the Newton direction $\boldsymbol{d}(\mu^k)$ at $\boldsymbol{x}^k$.

(3) *(Updating position and parameter)* Set $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \boldsymbol{d}(\mu^k)$ and $\mu^{k+1} \leftarrow \alpha\mu^k$ *(reducing the parameter).*

(4) Repeat (2)-(3) with $k \leftarrow k+1$.

In this process, we have $\{\mu^k\} \to 0$ and $\boldsymbol{x}^k$ remains close to $\boldsymbol{x}(\mu^k)$, allowing us to obtain fairly good approximations to $\lim_{\mu \to 0} \boldsymbol{x}(\mu)$ with direct computations of $\boldsymbol{x}(\mu)$ avoided. Graphically, it looks like the following, which suggests the iterates $\boldsymbol{x}^k$'s would travel in a "zigzag" trajectory that "follows the path".



6.3.7 **Steps in the primal path following algorithm.** Building upon this idea of approximating the central path by the Newton direction, we are now ready to describe the **primal path following algorithm** as follows.

(1) *(Inputs)* Specify the following inputs:

i. data of the standard form LP problem: A, $\boldsymbol{b}$, and $\boldsymbol{c}$, where A is assumed to have linearly independent rows

ii. the initial primal and dual feasible solutions: $\boldsymbol{x}^0$ and $(\boldsymbol{p}^0, \boldsymbol{s}^0)$, with $\boldsymbol{x}^0 > \boldsymbol{0}$ and $\boldsymbol{s}^0 > \boldsymbol{0}$

iii. the optimality tolerance $\varepsilon > 0$

iv. the initial barrier parameter μ^0 and the parameter $\alpha \in (0,1)$

(2) *(Initialization)* Set $k \leftarrow 0$.

(3) *(Optimality check)* If $(\boldsymbol{s}^k)^T\boldsymbol{x}^k < \varepsilon$, then stop and conclude that $\boldsymbol{x}^k$ and $\boldsymbol{p}^k$ are ε-optimal for the primal and dual respectively.

(4) *(Computing Newton direction)* Set:

- $X_k \leftarrow \operatorname{diag}\left(x_1^k, \ldots, x_n^k\right)$
- $\mu^{k+1} \leftarrow \alpha\mu^k$
- $\boldsymbol{p}^*, \boldsymbol{d}^* \leftarrow$ the solutions $\boldsymbol{p}, \boldsymbol{d}$ to the following system:

$$\begin{gathered}\mu^{k+1}(X_k^{-1})^2\boldsymbol{d} - A^T\boldsymbol{p} = \mu^{k+1}X_k^{-1}\boldsymbol{1} - \boldsymbol{c},\\ A\boldsymbol{d} = \boldsymbol{0}\end{gathered}$$

[Note: When solving problems, it is perhaps more practical to substitute the numerical values into the system and then solve it directly, rather than memorizing 🧠 the lengthy formulas in [6.3.5].]

(5) *(Updating solutions)* Set:

- $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^{k} + \boldsymbol{d}^{*}$
- $\boldsymbol{p}^{k+1} \leftarrow \boldsymbol{p}^{*}$
- $\boldsymbol{s}^{k+1} \leftarrow \boldsymbol{c} - A^{T}\boldsymbol{p}^{*}$

(6) *(Repeating prior steps)* Set $k \leftarrow k+1$ and go back to (3).

Remarks:

- *(Number of iterations needed)* The number of iterations taken for getting ε-optimal solutions is related to the choices of the inputs; see Bertsimas and Tsitsiklis (1997, Theorem 9.7).
- *(Performance)* It can be shown that this algorithm does run in polynomial time; see Bertsimas and Tsitsiklis (1997) for more details.
- *(Primal-dual path following algorithm)* The word "primal" in the "primal path following algorithm" indicates that the algorithm focuses on updating the solutions for the *primal* problem, and approximating central path in the *primal* feasible region. An extension is available, known as the *primal-dual path following algorithm*, which do these for both the primal and dual problems, and hence, as one may expect, exhibit an even better performance. Hence, the primal-dual path following algorithm is the standard interior point method used in many software due to its superior performance. However, we shall discuss it in details here; see Bertsimas and Tsitsiklis (1997, Section 9.5).

7 Integer Linear Programming

7.0.1 So far, we have focused on analyzing linear programming problems, where the decision variables are real-valued, corresponding to "continuous" optimization problems. However, in practice many variables cannot be modeled as real numbers, e.g., number of people must be integers, which bring in "discreteness" in optimization problems. This shows a limitation of the linear programming framework. To handle those problems where some variables can be integers, we need to consider another class of optimization problems, known as *integer linear programming (ILP) problems.*

7.1 Definition and Modelling Techniques

7.1.1 **Definition.** An **integer linear programming problem** can be expressed as

$$\begin{aligned} \text{min/max} \quad & f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} = c_1x_1 + \cdots + c_nx_n \\ \text{s.t.} \quad & \boldsymbol{a}_j^T\boldsymbol{x} \le / = / \ge b_j \text{ for all } j = 1, \dots, m, \\ & \text{some/all } x_j\text{'s are integers.} \end{aligned}$$

7.1.2 Of course, one clear and important application of ILP is to model linear optimization problems with some integer variables. But this is not the only reason for studying ILP. Another more *theoretical* motivation for developing ILP is that integer variables, while being "discrete", turn out to be helpful for modelling more complex "continuous" constraints also. Unfortunately, there is no general "rule" for how this can be done, unlike the procedure of reducing general LP problem to standard form as in [1.1.7]. Here, we will investigate one useful method for doing this, namely through the usage of *binary variables* to describe case-dependent constraints, e.g., "either-or constraints".

7.1.3 **Modelling constraints through binary variables.** A **binary variable** x is one that can only takes values 0 or 1. To include a binary variable x into an ILP problem, we can use the relationship $x \in \{0, 1\}$ iff x is integer and $0 \le x \le 1$, where the latter two constraints are permissible in an ILP problem. In view of this, sometimes we just write something like "$x \in \{0, 1\}$" in an ILP problem directly, which is understood to refer to the two constraints specified above.

To illustrate the usage of binary variables for modelling constraints, consider the following example of optimization problem involving an "either-or" constraint:

$$\begin{aligned} \min \quad & -3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \ge 2 \text{ or } x_1 + 2x_2 \ge 3 \\ & x_1, x_2 \ge 0 \end{aligned}$$

To deal with the "either-or" constraint $x_1 + x_2 \ge 2$ or $x_1 + 2x_2 \ge 3$, we can introduce a binary variable $y \in \{0, 1\}$ and express it as

$$\begin{aligned} x_1 + x_2 &\ge 2y \\ x_1 + 2x_2 &\ge 3(1 - y) \\ x_1, x_2 &\ge 0 \\ y &\in \{0, 1\}. \end{aligned}$$

The y's on the right-hand side of the inequalities serve as a "switch": If $y = 1$, then the inequality $x_1 + x_2 \ge 2$ must be satisfied, and the other inequality just becomes $x_1 + 2x_2 \ge 0$, which is always satisfied as $x_1, x_2 \ge 0$, so it does not affect the feasible region in such case. Similarly, if $y = 0$, then the inequality $x_1 + 2x_2 \ge 3$ must be satisfied, and the other inequality, $x_1 + x_2 \ge 0$ in this case, is always satisfied.

7.1.4 **Modelling disjunctive constraints.** In general, we can also deal with "either-or" constraints involving more than two inequalities *(disjunctive constraints)*. Suppose that we are given m constraints $\boldsymbol{a}_i^T\boldsymbol{x} \ge b_i, \quad i = 1, \dots, m$ and the sign constraint $\boldsymbol{x} \ge \boldsymbol{0}$, with $\boldsymbol{a}_i \ge \boldsymbol{0}$ for every i, and at least k of the

m constraints need to be satisfied. To model this disjunctive constraint, we can introduce m binary variables $y_1, \ldots, y_m$ and express it as

$$\begin{aligned} \boldsymbol{a}_i^T \boldsymbol{x} &\geq b_i y_i \quad \forall i = 1, \ldots, m, \\ \boldsymbol{x} &\geq \boldsymbol{0}, \\ \sum_{i=1}^{m} y_i &\geq k, \\ y_i &\in \{0, 1\} \quad \forall i = 1, \ldots, m \end{aligned}$$

Here, the y_i's serve as both "switches" and also count the number of inequalities satisfied, which is given by $\sum_{i=1}^{m} y_i$.

7.2 Cutting Plane Algorithm

7.2.1 After briefly studying how the ILP is useful for modelling constraints, we are then interested in knowing how ILP problems can be *solved*. Unfortunately, this is in general a difficult task, and algorithms for solving ILPs are often somewhat inefficient. Here, we will provide an overview of a relatively basic algorithm for solving ILPs, known as *cutting plane algorithm*, for dealing with ILPs where all variables are required to be integers:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \\ & \boldsymbol{x} \text{ integer.} \end{aligned}$$

[Note: Here "$\boldsymbol{x}$ integer" means that x_j is an integer for every $j = 1, \ldots, n$.]

The cutting plane algorithm is developed based on the observation that, if the LP problem with the integer variables constraint dropped *(relaxed problem)*,

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

results in an optimal solution with all variables being integers, then such solution would be also optimal for the ILP problem above.

7.2.2 **Steps in the cutting plane algorithm.** Consider the ILP

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \\ & \boldsymbol{x} \text{ integer.} \end{aligned}$$

with its LP relaxation

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

Suppose that the LP problem has a finite optimal value. The **cutting plane algorithm** is carried out as follows.

(1) *(Optimality check)* Find an optimal solution $\boldsymbol{x}^*$ to the LP problem. If $\boldsymbol{x}^*$ contains only integer variables, then conclude that $\boldsymbol{x}^*$ is optimal for the ILP problem.

[Note: The *dual simplex method* is often helpful for finding such optimal solution to the LP problem.]

(2) *(Finding a cutting plane)* If some entries of $\boldsymbol{x}^*$ are not integers, then find a **cutting plane** $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \geq b\}$ that separates $\boldsymbol{x}^*$ from all integer feasible solutions, which satisfies: (i) $\boldsymbol{a}^T\boldsymbol{x}^* < b$ and (ii) $\boldsymbol{a}^T\boldsymbol{x} \geq b$ for every integer feasible solution to the ILP.

[Note: It can be shown that such cutting plane always exists; see also [7.2.3] for a special case where a "formula" for cutting plane is available.]

(3) *(Adding constraint to the LP problem)* Add the constraint $\boldsymbol{a}^T\boldsymbol{x} \geq b$ to the LP problem, and go back to (1).

The basic idea is that, with the constraint $\boldsymbol{a}^T\boldsymbol{x} \geq b$ added to the LP problem, a new optimal solution to the LP problem results, and also no integer feasible solution is lost in the process by construction of the cutting plane. Then, intuitively, we would be able to get an integer optimal solution "eventually". Here, we shall not delve into the details about the theoretical justification of this method.

7.2.3 **Finding a cutting plane for optimal basic feasible solution.** To carry out the cutting plane algorithm in [7.2.2], we would need to find a cutting plane in each iteration, but there is not a general way to do that. Nevertheless, in the special case where the $\boldsymbol{x}^*$ found is an optimal *basic feasible solution* to the LP problem (which is the case if the dual simplex method is used to find the optimal solution), a cutting plane is given by $\boxed{\displaystyle\sum_{i\in I_N} x_i \geq 1}$ where I_N is the set of all nonbasic indices for $\boldsymbol{x}^*$ (whenever some entries in $\boldsymbol{x}^*$ are not integers).

Proof. First, we have $\sum_{i\in I_N} x_i^* = \sum_{i\in I_N} 0 = 0 < 1$, since every nonbasic variable is zero by definition. Next, fix any integer feasible solution $\boldsymbol{x} \geq \boldsymbol{0}$. Assume to the contrary that $\sum_{i\in I_N} x_i < 1$, which implies that $\sum_{i\in I_N} x_i = 0$ (as it is a nonnegative integer), i.e., $x_i = 0$ for all $i \in I_N$. But then this means that $B\boldsymbol{x}_B = B\boldsymbol{x}_B + N\boldsymbol{x}_N = A\boldsymbol{x} = \boldsymbol{b}$, and hence $\boldsymbol{x}_B = B^{-1}\boldsymbol{b} \overset{\text{(optimal basic feasible solution)}}{=} \boldsymbol{x}_B^*$. Therefore, we have $\boldsymbol{x} = \boldsymbol{x}^*$, suggesting that some entries in $\boldsymbol{x}$ are not integers, contradiction. □

References

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization.* Athena Scientific.

Concepts and Terminologies

Results

Section 1

- Proposition 1.1.a: equality of optimal values for two equivalent optimization problems
- Theorem 1.1.b: procedure for reducing any LP problem to an equivalent standard form LP
- Proposition 1.1.c: maximum of convex functions is still convex
- [1.1.9]: procedure for reducing optimization problems involving piecewise linear convex objective function and "$\leq$" constraints
- [1.2.6]: every polyhedron is convex
- Theorem 1.2.a: equivalence between vertex and basic feasible solution
- Theorem 1.2.c: specialized characterization of basic solutions for standard form polyhedra
- [1.2.12]: procedure for finding basic feasible solutions of standard form polyhedra
- Theorem 1.2.d: vertex exists iff a line is not contained
- Corollary 1.2.e: nonempty standard form polyhedron always has a vertex
- Theorem 1.2.f: finite optimal value of a LP problem is attained at a vertex
- Corollary 1.2.g: optimal solution of a LP problem always exists if the optimal value is finite

Section 2

- Proposition 2.1.a: a point is optimal iff $\boldsymbol{c}^T\boldsymbol{d} \geq 0$ for every feasible direction $\boldsymbol{d}$ at that point
- Proposition 2.1.b: expressions of feasible directions at basic feasible solutions
- Proposition 2.1.c: a basic feasible solution is optimal iff the reduced cost for every nonbasic variable is nonnegative, under nondegeneracy
- [2.1.5]: zero reduced cost of every basic variable
- Proposition 2.2.a: justification for change of basis in the simplex method
- [2.2.4]: procedure of the (non-revised) simplex method
- [2.2.6]: procedure of the revised simplex method
- [2.2.10]: procedure of the simplex method with tableau representation
- Proposition 2.3.a: original LP problem is feasible iff the optimal value of the auxiliary problem is 0
- [2.3.9]: two-phase algorithm
- [2.3.10]a: sufficient condition for optimal solution of the big-M method to satisfy $\boldsymbol{y} = \boldsymbol{0}$
- [2.3.10]b: obtaining optimal solution of the original LP problem from the optimal solution of the big-M problem
- [2.3.10]c: implication of having $-\infty$ as the optimal value of the big-M problem

Section 3

- [3.1.2]: the dual of the dual is the primal
- Theorem 3.1.a: relationship between duals of equivalent problems
- Theorem 3.2.a: weak duality
- Corollary 3.2.b: sufficient condition for infeasibility of the primal/dual problem
- Corollary 3.2.c: sufficient condition for having optimal solutions for the primal and the dual
- Theorem 3.2.d: strong duality
- [3.2.4]: possibilities for the primal and the dual
- Theorem 3.2.e: optimality condition based on complementary slackness
- Lemma 3.3.a: primal and dual feasibility implies optimality of basis matrix

Section 4

- Theorem 4.2.a: $\boldsymbol{b} \mapsto F(\boldsymbol{b})$ is convex
- [4.2.5]: $\boldsymbol{c} \mapsto G(\boldsymbol{c})$ is piecewise linear and concave
- [4.3.3]: $\theta \mapsto g(\theta)$ is piecewise affine and concave
- [4.3.4]: procedure for computing $g(\theta)$ by the simplex method

Section 5

- [5.1.2]: criterion for an iterative algorithm to run in polynomial time
- Proposition 5.1.a: simplex method takes exponential time
- [5.2.3]: properties of geometrical concepts related to the ellipsoid method
- Theorem 5.2.a: formula for the "next ellipsoid"
- [5.2.6]: implementation of the ellipsoid method
- Theorem 5.2.b: ellipsoid method solves linear feasibility problem correctly
- Proposition 5.2.c: vertices of polyhedron with integer data are bounded
- Proposition 5.2.d: a perturbation of polyhedron that preserves emptiness and makes nonempty polyhedron full-dimensional
- Proposition 5.2.e: lower bound for the volume of a bounded and full-dimensional polyhedron with integer data
- Theorem 5.2.f: linear feasibility problem with integer data can be solved in polynomial time
- [5.2.12]: applying the ellipsoid method to solve LP problem with integer data in polynomial time

Section 6

- Lemma 6.1.a: preliminary result for forming ellipsoids in interior
- Proposition 6.1.b: closed-form formula for optimal solution $\boldsymbol{d}^*$ to the problem of minimization over ellipsoid
- Proposition 6.1.c: optimal condition for "near optimality" for primal and dual
- Proposition 6.1.d: long-step variant with maximum norm (and also γ) leads to movements within interior
- Theorem 6.1.e: convergence of affine scaling algorithm
- Proposition 6.1.f: justification for the "big-M method" for initializing the affine scaling algorithm
- Lemma 6.2.a: lower bound on the potential function
- Theorem 6.2.b: number of iterations for getting ε-optimal solutions by potential reduction

Section 7

- [7.1.4]: method for modelling disjunctive constraints through binary variables
- [7.2.3]: formula of a cutting plane for the case where $\boldsymbol{x}^*$ is optimal basic feasible solution